

Knowledge Fusion Patterns for Design of Context-Aware Decision Support Systems

Alexander Smirnov and Tatiana Levashova

St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences, 39, 14th line, St. Petersburg, 199178, Russian Federation

smir@iias.spb.su, tatiana.levashova@iias.spb.su

Abstract. The objective of the research is to offer a methodology providing a feasible functionality of context-aware systems for the users. The systems are considered consisting of a set of knowledge sources. These sources are involved in knowledge fusion processes to support the systems functionalities. The core of the methodology is knowledge fusion patterns. They characterize knowledge sources involved in the knowledge fusion processes in terms of preservation/change of the sources' autonomies and structures. In the methodology, the patterns are used to specify the requirements to the knowledge sources from the systems to ensure the full systems functionalities. Matching the system's requirements against the user's requirements to the system functionality and the user's constraints on the sources of information and knowledge enables to offer feasible system functionality to the users.

Keywords: Context aware decision support, ontology-based context, knowledge fusion, knowledge fusion patterns, pattern-based methodology, emergency response.

1 Introduction

With the rapid development of ambient intelligence, pervasive and ubiquitous computing technologies the decision support systems are progressing toward becoming context aware. The importance of the knowledge fusion technology for the context-aware decision support systems (CADSSs) is reasoned by the intention of this technology. The objective of knowledge fusion is to integrate information and knowledge from multiple sources into some common knowledge that may be used for decision making and problem solving or may provide a better insight and understanding of the situation under consideration [1], [2], [3], i.e. knowledge fusion facilitates situation awareness and improves decision making. The main distinguishing feature of knowledge fusion lies in creation of a synergic effect from the integration of information and knowledge.

Methodologies for information system design and development distinguish different components comprising the systems. Processes, functions, people, information, and technology [4]; functions, resources, data, and material or immaterial outputs [5]; resources, context model, context processing engine/services, context storages, and context security rules/polices [6] are some examples of such components. The interactions of the systems' components determine the system behavior.

The objective of the present research is to offer a methodology providing a feasible functionality of CADSSs for the user. The CADSSs are considered consisting of a set of knowledge sources. In terms of the systems' components presented above, these sources are used

to represent system components (resources, people, contexts), provide system components (data, information, knowledge, functions with corresponding outputs), participate in system components (processes), etc. The system behavior and its functionality are determined by the interactions of the knowledge sources, which interact to share, exchange, and integrate/fuse the information/knowledge that they provide and to support the systems' functionalities.

The core of the methodology is knowledge fusion patterns [7]. The patterns were revealed based on the investigation of the knowledge fusion processes occurring in a CADSS for the emergency management domain. The contribution of the present paper is one of the possible pattern applications, that is the pattern-based methodology for design of CADSSs. A deep discussion about the patterns themselves goes beyond the scope of this paper.

The knowledge fusion patterns describe the knowledge fusion effects produced by the knowledge sources, specify states of the knowledge sources involved in knowledge fusion, and lift the effects at the ontology level. In the proposed methodology, the patterns are used to specify requirements to the knowledge sources from the system to ensure the system functionality. Generally, the set of knowledge fusion patterns includes patterns that can be used to specify the system's requirements ensuring the overall system functionality. Nevertheless, this work is limited to the patterns covering only the context-aware phase of the CADSS. It is the very phase where the system manifests its context-aware behavior.

The rest of the paper is as follows. The related research is briefly outlined in Section 2. Section 3 gives an overview of possible knowledge fusion results and introduces the elements of the language for the description of the knowledge fusion patterns. Section 4 presents the knowledge fusion results found in the CADSS for emergency management and provides specifications for the elements of the knowledge fusion patterns relevant to the discussion of the proposed methodology. The pattern-based methodology is presented in Section 5. The main results and limitations are discussed in the Conclusion.

2 Related Research

Methodologies for development of context-aware systems consider various classes of requirements important for the system design. They are the requirements to design any information system, the requirements of the classic distributed systems, the requirements of ubiquitous and pervasive systems, etc. This Section analyses only the requirements relevant to get the system context-aware. Such requirements specify the systems' needs in order that they could execute their context-aware functions, support context-aware decisions, and provide context-aware recommendations.

Some methodologies and approaches [8], [9], [10], [11] specify requirements to the tasks that the systems have to support to be context-aware. The systems, for instance, have to be able to represent, model, interpret, store, update context, be adaptable to the contexts, support reconfiguration mechanisms, etc.

Other approaches [12], [13], [14], [15] specify requirements regarding to the information relevant to the systems in order to they could fulfill their tasks. The needed information concern the locations, statuses (activities), and availabilities of the systems' resources including devices, services, users; the amount of possible users, the user preferences and other user characteristics (personal information, capabilities, and so on); the time, etc.

One more group of approaches [16], [17], [18] specifies requirements to the sources providing information in the systems. These requirements address adaptability of the operational states of the sources to the context, controllability of the sources' modalities, controllability of the sources' characteristics and locations, manageability of the sources, ability of the sources to cooperate, requirements to the access techniques, response time, etc.

The methodology proposed in this work specifies requirements to knowledge sources, which are the CADSS' constituents. The requirements address the autonomies and structures of the

sources. The knowledge sources get the CADSS context-aware due to their involvement in the knowledge fusion processes. The proposed methodology contributes to other existing methodologies for design and development of context-aware systems in the consideration of the knowledge fusion patterns for the specification of the requirements to the systems' resources and/or sources.

3 Knowledge Fusion Patterns

As it is said in the Introduction, knowledge fusion is characterized by creation of a synergic effect. That is, knowledge fusion results in appearance of a new knowledge. Based on the analysis of knowledge fusion studies, a number of knowledge fusion processes giving rise to different results have been distinguished:

1. Integration of multiple knowledge sources into a new knowledge object, which is a new knowledge source [19], [20].
2. Intelligent fusion of huge amounts of heterogeneous data / information from a wide range of distributed sources into a form which may be used by systems and humans as the foundation for problem solving and decision making [1], [2]. The intelligence assumes the consideration of the semantic contents of the sources being fused.
3. Integration of knowledge from various knowledge sources resulting in a completely different type of knowledge or new idea how to solve the problem [21], [22], or integration of different types of knowledge (domain, procedural, derived, presentation, etc.) resulting in a new knowledge type [3].
4. Inference of explicit knowledge from information/knowledge hidden in knowledge sources being integrated or fused [23].
5. Combining knowledge from different autonomous knowledge sources in different ways in different scenarios that results in discovery of new relations between the knowledge from different sources or/and between the entities this knowledge represents [24], [25].
6. Re-configuration of knowledge sources to achieve a new configuration with new capabilities or competencies or knowledge exchange to improve capabilities or competencies through learning, interactions, discussions, and practices [26].
7. Knowledge exchange to improve capabilities or competencies through learning, interactions, discussions, and practices [26].
8. Involving knowledge from various sources in problem solving, which results in a solution [27].

From the analysis above it is noticed that different processes can produce the same results, and different results may be outcomes of the same process. The following not mutually exclusive kinds of new knowledge produced as the knowledge fusion results are distinguished:

- a new knowledge source (the result of the process 1);
- a new knowledge created from data/information (the result of the process 2);
- a new type of knowledge (the result of the process 3);
- a new problem solving method or idea how to solve the problem (the result of the process 3);
- a new knowledge about the conceptual scheme (new relations, concepts, properties, etc. (the result of the processes 4 and 5);
- new capabilities/competencies of a knowledge object (an object that produces or contains knowledge) (the result of the processes 6 and 7);
- a solution for the problem (the result of the process 8).

Knowledge fusion involves multiple sources in the integration processes. Autonomies and structures of such sources have been chosen as the concepts in terms of which knowledge fusion patterns are revealed. In the context-aware systems integration of information/knowledge refers to the process of integration of their conceptual structures. Therefore, source's structure is an

obligatory concept taken into account by the integration. In this research, by sources' structures the conceptual structures representing the knowledge in the knowledge sources are meant.

Autonomy creates awareness of the reliability of information/knowledge represented in the sources. The consideration of the knowledge source autonomy concept is reduced to the consideration of relations existing between knowledge sources regardless of their structures. Autonomous knowledge source is an independent source having no relationships with other sources. Such source can get changed at any time, at that, the changes in this source produce no changes in other sources. On the contrary, non-autonomous source is linked to other (non-autonomous) sources. Changes in a non-autonomous source produce appropriate changes in the related sources.

CADSSs operate in dynamic environments. Information and knowledge represented in the environmental sources that are related to the internal system sources (i.e., the environmental sources and system ones are non-autonomous) are considered to be more reliable than information/knowledge represented in the autonomous environmental sources. An argument in favor of this is any changes in the linked (non-autonomous) environmental sources are reflected in the system sources.

The patterns measure knowledge fusion outcomes in terms of preservation/change of the structures and autonomies of the initial and target knowledge sources, and in terms of the results the knowledge fusion processes produce in the system.

Initial knowledge sources are the sources that are integrated leading to the emergence of a new knowledge (producing some knowledge fusion result). The sources resulting from the knowledge fusion or enclosing the knowledge fusion result are referred to as target knowledge sources.

The knowledge fusion patterns are described using a pattern description language [7]. The detailed presentation of the patterns elements is as follows:

Name: a name to refer to the pattern

Problem: a problem the knowledge fusion process solves

Solution: a meaningful description of the knowledge fusion process

Initial knowledge source(s): knowledge source(s) that are integrated leading to producing some knowledge fusion result

Target knowledge source(s): knowledge source(s) resulting from the knowledge fusion or enclosing the knowledge fusion result

Related pattern (may be omitted): an alternative pattern that can be used instead of the described one or in parallel or after termination of the described pattern

Exception (may be omitted): a description of cases when the pattern is not applicable

Autonomy pre-states: the degree of autonomy of knowledge sources before the knowledge fusion process. Three degrees are provided for: autonomous, non-autonomous, and n/a (for a non-existing knowledge source)

Result in the system: the result the knowledge fusion process produces in the CADSS

Result in ontology terms: ontology-based generalization of the result produced

Post-states: the degrees of preservation of knowledge sources autonomies and structures after the knowledge fusion process completes. For the knowledge sources autonomies the degrees introduced in the pre-state descriptions are kept on. Three degrees of preservation of the knowledge sources structures are provided for: preserved, changed, and new (for a new knowledge source)

Schematic representation: the knowledge fusion process represented schematically

Stage: the stage of the CADSS scenario where the knowledge fusion process occurs.

In this work, the patterns usage in the pattern-based methodology for design of CADSSs is offered. Prior to discuss this methodology, the CADSS for the emergency management domain is investigated for presence of the knowledge fusion results in its scenario and the pattern elements that are relevant to the discussion of the methodology are introduced.

4 Context-Aware Decision Support System for Emergency Management

The CADSS for emergency management is intended to support decisions on planning emergency response actions. The system scenario follows two main phases: preliminary and executive. At the preliminary phase an application ontology, which describes knowledge of the emergency management domain, is built. This ontology specifies knowledge to describe the emergency situations happening in this domain along with the problems requiring solutions in these situations. The application ontology is a knowledge source fusing two types of knowledge: domain and problem-solving. The executive phase concerns context-aware support of the decision maker with alternative decisions, decision implementation, and archiving. The focus of this paper is the executive phase.

4.1 Knowledge Fusion Results

A two-level representation of emergency situation is used in the CADSS. At the first level the situation is represented by *abstract context* that is an ontology-based intentional model of the situation. The abstract context represents knowledge relevant to support decisions in the current emergency situation. This knowledge is captured from the application ontology. The result of the capturing is unrelated slices of the application knowledge. The abstract context fuses these slices into one knowledge piece. As well as the application ontology the abstract context represents knowledge of two types. Figure 1 depicts an example of the abstract context built for the fire situation. Referring to the domain knowledge, this abstract context, along with other issues, specifies that services provided by emergency teams and fire brigades are required in the fire situation. These teams and brigades can use ambulances, fire engines and special-purpose helicopters for transportation. The problem-solving knowledge is collapsed in the “Emergency response” class. This class specifies subproblems to solve the problem of planning fire response actions and methods for the problems solving. Some examples of subproblems are the routing problem, the problems of search for and selection of available emergency teams and firefighter brigades.

The knowledge fusion result produced at the stage of the abstract context creation is a new knowledge source (the abstract context) of the same type as the initial knowledge source (the application ontology). The application ontology preserves its structure and autonomy; the abstract context becomes an autonomous knowledge source with a proper structure. The process of the abstract context creation is generalized by the *simple fusion* pattern (Table 1). Although in the table the application ontology is indicated as the single initial knowledge source, the target knowledge source is the result of fusion since it fuses multiple pieces of knowledge captured from this ontology.

Table 1. Simple fusion

Pattern element	Initial knowledge source	Target knowledge source
Knowledge source	application ontology	abstract context
Autonomy pre-state	autonomous	n/a
Structure post-state	preserved	new
Autonomy post-state	autonomous	autonomous



Figure 1. Abstract context for fire situation: concept view (a fragment)

In the abstract context the captured knowledge may result in discovery of new relationships between the knowledge unrelated in the application ontology. These relationships are the result of deductive inference. Generally, any kind of knowledge representation items can be inferred. The inferred items are considered as a knowledge fusion result that is a new knowledge. As this result is a consequence of the integration of multiple pieces of the application knowledge, it is considered as the result of knowledge fusion. The inferred knowledge is introduced in the abstract context, hereby changing its structure. In the case of the abstract context refinement the abstract context plays the role of the initial and target knowledge source at the same time. The process of the abstract context refinement is generalized by the *extension* pattern (Table 2).

Table 2. Extension

Pattern element	Initial knowledge source	Target knowledge source
Knowledge source	abstract context	abstract context
Autonomy pre-state	autonomous	autonomous
Structure post-state	changed	changed
Autonomy post-state	autonomous	autonomous

At the second level the emergency situation is represented by operational context that is an instantiation of the abstract context for the actual circumstances. The operational context is produced through the semantic fusion of data/information from multiple environmental resources within the ontological structure of the abstract context. For this the resources organize a resource network. Nodes of this network represent the resources; network arcs signify the order of the nodes execution.

Firstly the operational context is a copy of the abstract context with structural elements for the representation for values of the concepts properties or with default values in these elements. As soon as the resources start instantiating this copy, they lose their autonomies. When all the properties have acquired values, i.e. the concepts have become instantiated, the operational context is produced. At that, information from the resources is constantly coming into this context. The operational context and the environmental resources are related over the period of decision making and decision implementation. In practice, the operational context represents the map of the area around the emergency event where the situation dynamic is represented (the mobile responders are moving, the traffic situation is changing, etc.) (Figure 2).

The knowledge fusion results obtained at the stage of the operational context producing are 1) the operational context is a new knowledge source; 2) this context is a knowledge source created from data/information; and 3) the operational context represents knowledge of a new dynamic type. The abstract context preserves its structure and autonomy when the operational context is produced. The operational context is a new non-autonomous knowledge source. The instantiated fusion pattern (Table 3) generalizes the process of the operational context producing. As it is said above, the operational context fuses the information from the environmental resources. Nevertheless, the abstract context is considered as the initial knowledge source in the pattern. The reason of this is that the abstract context is the primary basis for the operational context.

Table 3. Instantiated fusion

Pattern element	Initial knowledge source	Target knowledge source
Knowledge source	abstract context	operational context
Autonomy pre-state	autonomous	n/a
Structure post-state	preserved	new
Autonomy post-state	autonomous	non-autonomous

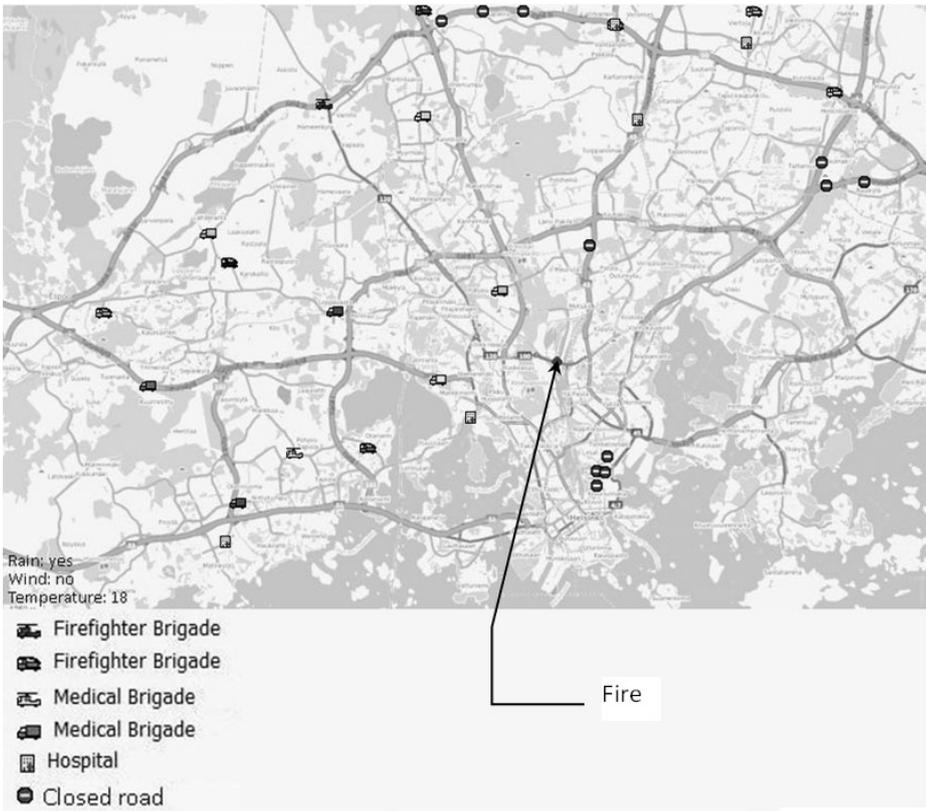


Figure 2. Operational context: new knowledge created from data/information

The abstract contexts are reusable components of the CADSS. The reuse of an abstract context in settings when the available resources are not intended to solve the problems specified in this context may result in finding alternative resources. For instance, one unavailable method can be substituted for a sequence of methods providing by the available resources. This leads to a new configuration of the resource network.

Figure 3 illustrates the case when the abstract context specifies the routing problem as a hierarchy of methods one of which ('GetLocation') returns the current locations of objects in the format of point coordinates on the map. The example under consideration requires to determine the locations of hospitals. The method 'GetLocation' uses data from sensors.

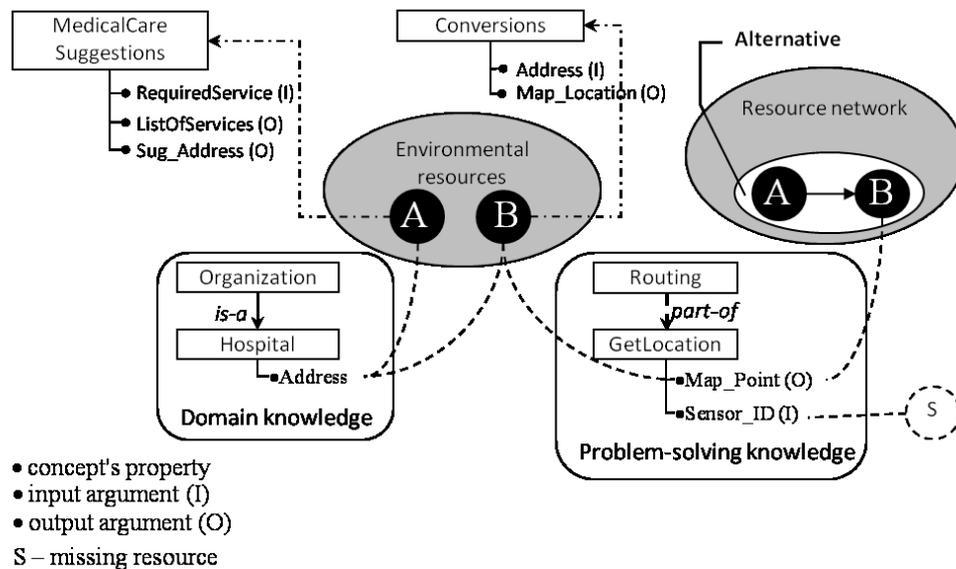


Figure 3. Discovery of alternative problem-solving methods

The set of environmental resources comprises no sensors dealing with static objects like hospitals, and this set comprises some other resources. One of them (A) implements a method ('MedicalCareSuggestions') intended to make recommendations what medical care organizations can be used to access some specific medical service. This source contains a database with information about the hospitals. The resource A returns the hospitals' addresses in an address format. The other resource (B) implements the method ('Conversions') that converts the address formats into the format of coordinates. The successive execution of the methods 'MedicalCareSuggestions' and 'Conversion's is an alternative way to calculate the hospital locations in the format of coordinates.

At the stage of the abstract context reuse the knowledge fusion result is twofold: a new (alternative) problem solving method and a new configuration of the resource network. A specification of the new method(s) is introduced in the abstract context. The context structure is changed, at that this context remains autonomous. The autonomies and structures of the resources representing the new method(s) are preserved.

In the case of abstract context reuse the knowledge fusion pattern is nested. The main pattern configured fusion (Table 4) includes the extension pattern. The extension pattern corresponds to the introducing the new specification in the context.

Table 4. Configured fusion (the main pattern)

Pattern element	Initial knowledge source	Target knowledge source
Knowledge source	abstract context	resource network
Autonomy pre-state	autonomous	autonomous
Structure post-state	changed	preserved
Autonomy post-state	autonomous	autonomous

Besides producing the operational context, the environmental resources solve the problem of planning emergency response actions based on this context. This problem is solved as a constraint satisfaction problem. The solution is a set of alternative emergency response actions feasible in the current emergency situation. An emergency response plan is a set of emergency responders with required helping services, schedules for the responders' activities, and transportation routes for the mobile responders. The types of emergency responders being included in the plans depend on their availabilities, the emergency situation type, the geography and infrastructure of the region. Figure 4 introduces an example of a response plan for the fire situation. In the plans earlier independent entities become related, i.e. new relations between these entities have arisen.

The plans are represented in the picture of the operational context. In this way, the operational context and the results of problem solving are fused forming, at that, a new knowledge source. This new source represents knowledge of a new type (the instantiated knowledge fused with the solution set).

During problem solving the operational context dissolves within the new knowledge source and does not preserve the structure and autonomy. At time of alternatives generation the environmental resources and the operational context are related (non-autonomous). The knowledge fusion effects produced at this stage are 1) new relations between entities, 2) a problem solution, and 3) a new knowledge source of a new type. Table 5 presents a fragment of the *flat fusion* pattern for the problem solving processes.

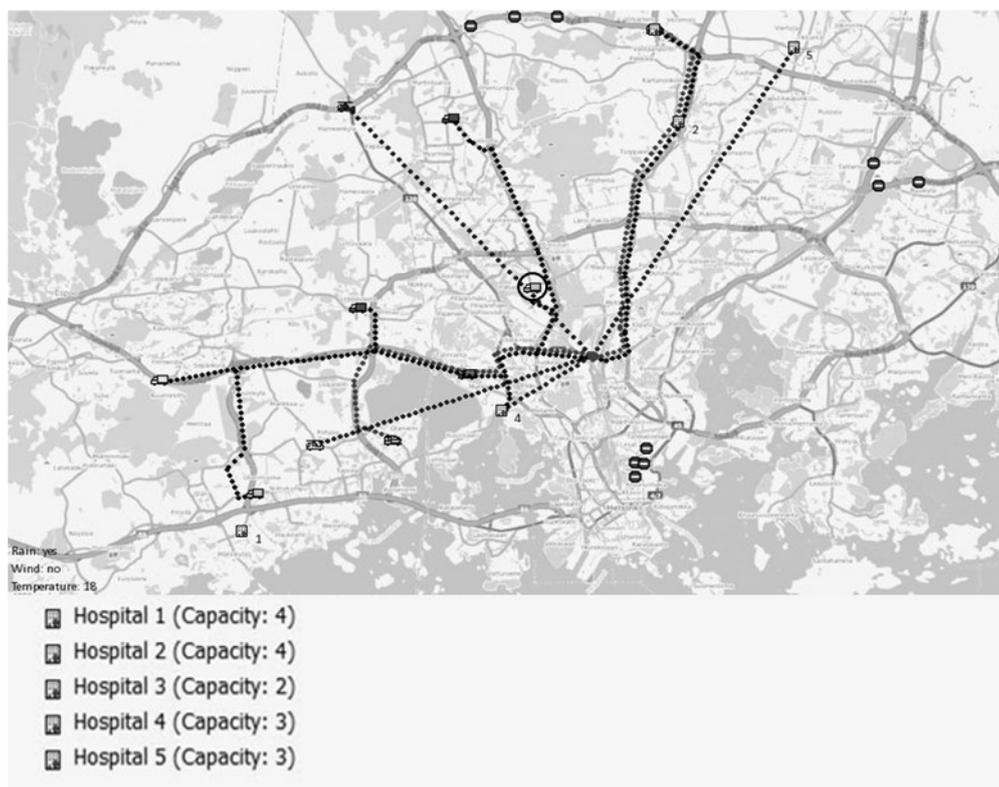


Figure 4. Response actions plan

Table 5. Flat fusion

Pattern element	Initial knowledge source	Target knowledge source
Knowledge source	operational context	knowledge source fusing the operational context and the set of alternatives
Autonomy pre-state	non-autonomous	n/a
Structure post-state	changed	n/a
Autonomy post-state	n/a	autonomous

The decision maker chooses an alternative from the set of feasible ones. The system facilitates the process of decision making by providing efficiency criteria which the decision maker could apply to the proposed set of alternatives. Applying such criteria allows the CADSS to narrow down the set of alternatives to 1 – 3 ones. The chosen alternative is considered as the decision. According to this decision the response actions are undertaken.

The situation may change from the moment the decision was made to the moment of its implementation. The responders whom the decision is delivered may be unable to implement it in the changed circumstances. In some cases, the activities assigned to the responders who become unable to operate can be delegated to or redistributed between other responders participating in the decision implementation. As a result of this, the responders that are ready to take the assignments gain new capabilities / competencies.

For instance, an emergency team trained to rescue operations has failed in the course of actions because of a road destruction, ambulance blockage, etc. In certain cases these operations can be delegated to available teams. In the CADSS the emergency responders are represented by their profiles. In the case of consent, the plan (the decision) is adjusted accordingly and the profiles of the teams agreed to take part in the rescue operations are extended with the new capability.

At the time of decision implementation, the responders taking part in the response plan are not autonomous. Moreover, in the course of the response actions the structures of their profiles as well as the decision structure may change. The changed decision structure results in changing the structure of the knowledge source containing the set of solutions. This knowledge source is not autonomous until the decision is implemented. The knowledge fusion result produced in the course of decision implementation consists in gaining new capabilities / competencies by the emergency responders. The *adaptation* pattern (Table 6) generalizes this case.

Table 6. Adaptation

Pattern element	Initial knowledge sources	Target knowledge sources
Knowledge source	<ul style="list-style-type: none"> • knowledge source representing the decision • profiles of the emergency responders 	
Autonomy pre-state	non-autonomous	non-autonomous
Structure post-state	Changed	changed
Autonomy post-state	non-autonomous	non-autonomous

Once the interactions of the decision maker with the CADSS have been finished (the decision has been implemented), the abstract context, the operational context, the decision, and the resources' representations are saved in a context archive. At that, the operational context and the resources' representations are saved in their states at the instant of the alternatives generation. In the archive all the listed components are related (non-autonomous). The archive is used to an inference of new knowledge based on the accumulated one.

For instance, the emergency team encircled in Figure 4 participated in different emergency response actions. Some operational contexts in which this team appeared and then participated in

corresponding actions do not represent any instances of the class 'Emergency response organization' specified in the abstract context. This assumes that the emergency team in question is a part of one of the hospitals represented in the operational contexts together with this team. Based on the operational context (Figure 5) it can be concluded that most probably the team is a part of hospital 5 represented in this context since the context does not represent any other hospitals from Figure 4 except for this one. The *Part-of* relationship between hospital 5 and the encircled emergency team is the newly revealed relation. The revealing of a new knowledge based on a set of observations is a kind of inductive inference.

The inference produces a new knowledge about conceptual schemes as the knowledge fusion result. This new knowledge is specified in the application ontology. As a result of this, the structure of the application ontology is changed but its autonomy is preserved. Table 7 shows a fragment of the *historical fusion* pattern for the process of archival knowledge management.

Table 7. Historical fusion

Pattern element	Initial knowledge sources	Target knowledge source
Knowledge source	operational contexts	application ontology
Autonomy pre-state	non-autonomous	autonomous
Structure post-state	Preserved	changed
Autonomy post-state	non-autonomous	autonomous



Figure 5. History for an emergency team

4.2 Pattern Relations

The relations between the knowledge fusion patterns in the CADSS are shown in Figure 6 (the fire response scenario is used as the example).

The abstract context is created using the simple fusion pattern. The application ontology serves as the pattern input. The abstract context represents non-instantiated ontological

knowledge for solving the problem of planning fire response actions. This context is the input for three patterns.

The first one is the extension pattern. This pattern is applied when the knowledge represented in the abstract context can be used to infer a new knowledge. In the scenario under consideration the new relationship between the representations of a mobile responder and the routing method is inferred. In the extension pattern the abstract context is both the pattern's input and output.

The second pattern, which uses the abstract context as its input, is the configured fusion. This pattern is applied when the abstract context is reused, i.e. when this context is put into a situation with a new (earlier unused) set of environmental sources. These sources are expected to instantiate the abstract context and solve problems specified in it. To fulfill this expectation, a new set of environmental sources has to configure a new knowledge source network. In the fire response scenario the abstract context is put in the situation in which the knowledge source (a sensor) intended to provide the hospitals' locations in the format of coordinates is missing. But the set of environmental sources includes two other knowledge sources. These sources provide the methods 'MedicalCareSuggestions' and 'Conversions'. Successive execution of the two available methods creates an alternative to the missing source. The two methods get specified in the abstract context. That is, the abstract context is both the input and output of the configured fusion pattern.

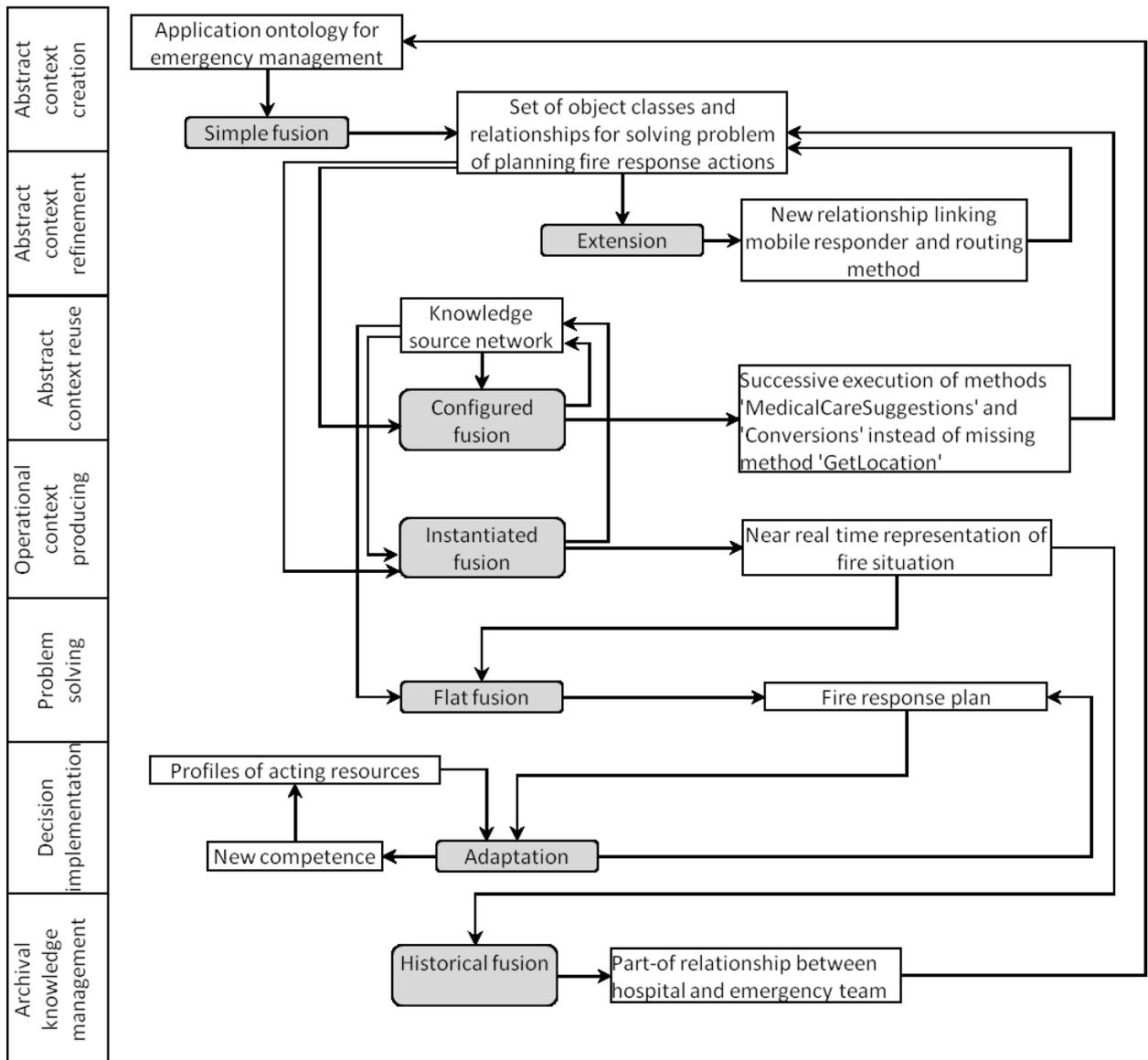


Figure 6. Usage of knowledge fusion patterns for planning fire response actions

The third pattern, in which the abstract context serves as the pattern's input, is the instantiated fusion. This pattern is used to produce an operational context that is a near real time representation of the fire situation. The knowledge source network configured according to the abstract context produces the operational context. The operational context is the output of the instantiated fusion pattern.

Besides the operational context is a near real time representation of the situation, it specifies the problem of planning fire response actions. The specification indicates what data from the situation representation are to be used to solve this problem. The operational context is the input for the flat fusion pattern. This pattern is responsible for problem solving. The result of problem solving is the set of fire response plans. The output of the flat fusion pattern is a knowledge source fusing the near real time representation of the fire situation and the set of plans. After the decision maker has chosen one plan, this knowledge source fuses the near real time representation of the fire situation and the chosen plan.

The source representing the fire situation along with the fire response plan is an input for the adaptation pattern. The adaptation pattern is applied when one of the fire responders becomes unable to participate in the planned fire response actions and the activities planned to this responder can be delegated to other responders from the plan. If a responder agrees to undertake the delegated activities the plan is corrected appropriately, the corrections get introduced in the knowledge source representing this plan. The profile of this responder is extended with the new competence corresponding to the undertaken activity. The profiles of the fire responders included in the plan are other inputs for the adaptation pattern. The outputs of this pattern coincide with the pattern's inputs (the source representing the fire situation along with the fire response plan and the responders' profiles).

The operational contexts or near real time representations of different emergency situations the CADSS has ever dealt with are saved in the archive. The accumulated operational contexts provide the basis for inductive inference of a new explicit knowledge from the hidden one. This is the case of the historical fusion pattern usage. The input of this pattern is the set of accumulated operational contexts. It gets the new knowledge introduced in the application ontology, which is the output of this pattern. In the fire response scenario a new (not specified in the application ontology) relationship between the hospital and the emergency team is inferred. This relationship indicates that the emergency team participated in different emergency response operations belongs to the hospital found together with this team is the operational contexts.

5 Pattern-Based Methodology

The offered methodology follows four steps: 1) capturing the user requirements to the system functionality; 2) specification of the system's information requirements; 3) matching the system requirements against the user requirements; 4) finding feasible system functionality.

The proposed patterns enable to formulate the system requirements in terms of the patterns' inputs/outputs (Table 8). Further on in this paper only the patterns' inputs are taken into account to formulate the requirements. The patterns' inputs and outputs jointly are planned to be used to track the information flows across the CADSS' stages and between the patterns. These flows manifest interrelationships between the stages and thereupon allow ones to specify explicitly in the methodology what output scenario components can serve as the input at what stage.

Table 8. System requirements

Stage	Input	Output	Functionality	Pattern
Abstract context creation	Autonomous application ontology	Autonomous abstract context	Creation of a non-instantiated ontology-based model of the situation	Simple fusion
Abstract context refinement	Autonomous modifiable abstract context	Autonomous abstract context	Inference of new (contextual) knowledge	Extension
Abstract context reuse	<ul style="list-style-type: none"> Autonomous modifiable abstract context Autonomous reconfigurable resource network 	<ul style="list-style-type: none"> Autonomous abstract context Autonomous resource network 	Reconfiguration of the resource network according to the current circumstances	Configured fusion
Operational context producing	<ul style="list-style-type: none"> Autonomous abstract context Resources able to lose their autonomies 	<ul style="list-style-type: none"> Non-autonomous modifiable operational context Non-autonomous resource network 	Creation of a near real-time picture of the situation	Instantiated fusion
Problem solving	<ul style="list-style-type: none"> Non-autonomous modifiable operational context Resources able to lose their autonomies 	<ul style="list-style-type: none"> New autonomous knowledge source Autonomous resource network 	Providing the decision maker with a set of alternative decisions	Flat fusion
Decision implementation	<ul style="list-style-type: none"> Non-autonomous modifiable knowledge source representing the decision Non-autonomous modifiable actors' profiles 		Gaining new capabilities / competencies by actors	Adaptation
Archival context management	<ul style="list-style-type: none"> Non-autonomous operational contexts Autonomous modifiable application ontology 	Autonomous application ontology	Inductive inference of new knowledge	Historical fusion

The analysis of the applicable patterns results in the following system functionality. The system can reuse the existing abstract context. As introducing any knowledge into this context is not allowed, this context can be reused if the previously used set of environmental resources is available. At that, any new network configurations are senseless as they cannot be specified in the context. Having the abstract context and the resource network the system can provide the user with an operational context (the dynamic picture of the situation) and a set of feasible decisions in this situation. As the user is able to manage the actors' profiles, the system provides him/her with the ability to manage the decision implementation. The operational contexts produced based on the reusable abstract context can be archived, inductive inference over them can be supported but the inference results cannot be retained.

The system requirements (Table 8) are formulated for the particular CADSS. The patterns' parameters are presented in the way they are used in this CADSS. These parameters generalized are presented in Table 9 (the columns "System requirements"). These columns formalize possible parameters' values for general cases and formulate general conditions for the patterns applicability. In the table the following notation is used: "a" – autonomous knowledge objects, "na" – non- autonomous, "m" – modifiable, "nm" – not-modifiable, "/" – logical OR, "&" – logical AND. The resource network is considered as a single knowledge object. Modifiable resource network means that this network is reconfigurable, at that any changes inside the network nodes are not supposed (the structures of the resource organizing the network are not changed).

Table 9. Pattern-based requirements

Pattern	System requirements		User requirements	Pattern applicability
	Input(s)	Parameter value	Parameter value	
Simple fusion	Application ontology	a/na/m/nm	unavailable	n/a
Extension	Abstract context	a/na&m	nm	n/a
Configured fusion	Abstract context	a/na&m	nm	n/a
	Resource network	a/na/m/nm	a/na&m	applicable
Instantiated fusion	Abstract context	a/na/m/nm	nm	applicable
	Resource network	na&m/nm	a/na&m	applicable
Flat fusion	Operational context	na&m	not defined = na&m	applicable
	Resource network	na&m/nm	a/na&m	applicable
Adaptation	Decision	na&m	not defined = na&m	applicable
	Actors' profiles	na&m	na/a&m	applicable
Historical fusion	Operational contexts	a/na/m	not defined = a/na/m	applicable
	Application ontology	a/na&m	unavailable	n/a

User requirements are demonstrated by an example. For instance, the user does not possess any application ontology. Though, he/she has an unalterable abstract context representing the abstract situation this user usually deals with and a set of resources authorized for his/her needs. As well, this user can manage the actors' profiles as at he/she discretion. The example of the user requirements is presented in the column "User requirements" (Table 9). "Not defined" parameter value means that the user has no specific requirements to the input component. The system designers can manipulate such a component according to their choice. Here it is supposed that the system provides the user with the components resulted in the system scenario execution with the parameters required for the overall system functionality.

The system requirements are matched against the user requirements to determine the patterns applicability for this user (the column "Pattern applicability" of Table 9). Table 9 demonstrates that the fully applicable patterns are *instantiated fusion*, *flat fusion*, and *adaptation*. The *configured fusion* and *historical fusion* patterns are applicable partly. The applicability of the *configured fusion* covers management of the resource network. The *historical fusion* pattern allows for some inductive inference. The remaining patterns are inapplicable.

The proposed functionality means that the user can deal with the situation he/she usually deals with, solve repetitive problems with different values for the problems' variables, and manage the decision implementation. For instance, applying to the emergency management domain, the system can support decisions on the emergency situations caused by one and the same type of event. At that, such types of events are supposed to happen in some area where the system has access to a fixed and the same set of environmental resources.

Conclusion

A pattern-based methodology for design of context-aware decision support systems has been proposed. Such systems are supposed to support knowledge fusion processes. The core of the methodology is knowledge fusion patterns. The patterns, along with other issues, characterize knowledge sources involved in the knowledge fusion processes in terms of preservation/change of the sources' autonomies and structures. In the methodology, the patterns are used to specify the requirements to the knowledge sources from the system to ensure the full system functionality.

The proposed methodology enables to offer feasible system functionality to the users depending on their requirements to the functionality and the constraints on the sources of information and knowledge. The methodology contributes to existing methodologies and

approaches to system design and development in the consideration of the knowledge fusion patterns for the specification of the requirements to the systems' resources and/or sources.

So far, the proposed methodology is limited to the requirements to the sources providing information and knowledge for knowledge fusion. In perspective it is planned to specify requirements to the sources that contain the knowledge fusion results, i.e., the target sources. This will allow the designers to be more specific about the feasible system functionality due to consideration of the sources that are not specified explicitly as target ones, but contain information and knowledge required at certain stage of system usage. As well, the analysis of requirements to the initial sources and target ones jointly will enable to define dependencies between the internal knowledge sources (the application ontology, the abstract contexts, and the archived operational contexts) and the system functionality and to make decisions if a certain pattern is needed in the current scenario. For instance, usage of the adaptation pattern and the historical fusion pattern is not necessary in all decision situations.

Acknowledgements

The present research was supported partly by projects funded by grants 13-07-12095, 13-07-13159, 14-07-00345, 14-07-00427 of the Russian Foundation for Basic Research, the project 213 of the research program “Information, control, and intelligent technologies & systems” of the Russian Academy of Sciences (RAS), and the project 2.2 of the Nano- & Information Technologies Branch of RAS.

References

- [1] P. Alun, K. Hui, A. Gray, P. Marti, T. Bench-Capon, Z. Cui, and D. Jones, “Kraft: an Agent Architecture for Knowledge Fusion,” *Int. J. Coop. Inf. Syst.* 10 (1–2), 2001, pp. 171–195.
- [2] R. Scherl and D. L. Ulery. Technologies for Army Knowledge Fusion. Final report no. ARL-TR-3279, Computer Science Department, Monmouth University, Monmouth, 2004.
- [3] C.W. Holsapple and A.B. Whinston, “Building Blocks for Decision Support Systems,” in G. Ariav, J. Clifford, Eds., *New Directions for Database Systems*. Ablex Publishing Corp., Norwood, 1986, pp. 66–86.
- [4] S. Alter, “Defining Information Systems as Work Systems,” *Eur. J. Inform. Syst.* 17, 2008, pp. 448–469. Available: <http://dx.doi.org/10.1057/ejis.2008.37>
- [5] P. Bernus, *Handbook on Architectures of Information Systems*. New York, NY: Springer 2006.
- [6] K. Henriksen, J. Indulska, T. McFadden, and S. Balasubramaniam, “Middleware for Distributed Context-Aware Systems,” in R. Meersman, Z. Tari, Eds., *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE. LNCS*, vol. 3760, 2005, pp. 846–863. Available: http://dx.doi.org/10.1007/11575771_53
- [7] A. Smirnov, T. Levashova, and N. Shilov. Patterns for Context-Based Knowledge Fusion in Decision Support. Information Fusion. 2013. Available: <http://dx.doi.org/10.1016/j.inffus.2013.10.010>
- [8] H. Chen, T. Finin, and A. Joshi, “Semantic Web in the Context Broker Architecture,” in *Proceedings of the 2nd IEEE International Conference on Pervasive Computing and Communications (PerCom04)*. Orlando, FL, 2004, pp. 277–286. Available: <http://dx.doi.org/10.1109/PERCOM.2004.1276865>
- [9] J. Malek, M. Laroussi, and A. Derycke, “A Multi-Layer Ubiquitous Middleware for Bijective Adaptation between Context and Activity in a Mobile and Collaborative Learning,” in *International Conference on Systems and Networks Communications (ICSNC'06)*. Tahiti, French Polynesia. 2006. Available: <http://dx.doi.org/10.1109/ICSNC.2006.4>

- [10] D. Zhang, T. Gu and X. Wang, "Enabling Context-Aware Smart Home with Semantic Web Technologies," *International Journal of Human-friendly Welfare Robotic Systems*, 2005.
- [11] M. Hussein, J. Han, and A. Colman, "An Architecture-based Approach to Context-Aware Adaptive Software Systems," Tech. Rep.: C3-516_04. Faculty of Information and Communication Technologies, Swinburne University of Technology, Melbourne Australia, 2011.
- [12] A. K. Dey and G. D. Abowd, "A Conceptual Framework and a Toolkit for Supporting Rapid Prototyping of Context-Aware Applications," *Journal Human-Computer Interactions (HCI)*, 16 (2-4), 2001, pp. 97-166. Available: http://dx.doi.org/10.1207/S15327051HCI16234_02
- [13] J. Indulska and P. Sutton, "Location management in Pervasive Systems," in *Proceedings of the Australasian Information Security Workshop Conference on ACSW Frontiers 2003*, vol. 21, 2003, pp. 143-151.
- [14] D. Hong, D. K. W. Chiu and V. Y. Shen, "Requirements Elicitation for the Design of Context-Aware Applications in a Ubiquitous Environment," in *Proceedings of the 7th International Conference on Electronic Commerce (ICEC'05)*. ACM. New York, NY, USA, 2005, pp. 590-596. Available: <http://dx.doi.org/10.1145/1089551.1089658>
- [15] M. Baldauf, S. Dustdar, and F. Rosenberg, "A Survey on Context-Aware Systems," *International Journal of Ad Hoc and Ubiquitous Computing*, 2 (4), 2007, pp. 263-277. Available: <http://dx.doi.org/10.1504/IJAHUC.2007.014070>
- [16] C. Allison, M. Bramley, R. Michaelson, and J. Serrano, "An Integrated Framework for Distributed Learning Environments," in *Advances in Concurrent Engineering: Proceedings of 6th ISPE International Conference on Concurrent Engineering*, 1999, pp. 345-354.
- [17] A. Wood et al. *Context-Aware Wireless Sensor Networks for Assisted Living and Residential Monitoring*. Network, 22 (4), 2008, pp. 26-33. <http://dx.doi.org/10.1109/MNET.2008.4579768>
- [18] J. Rubio-Loyola, "Manageability of Future Internet Virtual Networks from a Practical Viewpoint," in G. Tselentis, Eds., *Towards the Future Internet*. IOS Press, 2010, pp. 105-114. Available: <http://dx.doi.org/10.3233/978-1-60750-539-6-105>
- [19] T.-T. Kuo, S.-S. Tseng, and Y.-T. Lin, "Ontology-Based Knowledge Fusion Framework Using Graph Partitioning," In: P.W.H. Chung, C.J. Hinde, M. Ali, Eds., *IEA/AIE 2003*. LNAI, vol. 2718, 2003, pp. 11-20.
- [20] J. Gou, J. Yang, and Q. Chen, "Evolution and Evaluation in Knowledge Fusion System," In: J. Mira, J. R. Alvarez, Eds., *IWINAC 2005*. LNCS, vol. 3562, 2005, pp. 192-201.
- [21] K.-R. Lee, "Patterns and Processes of Contemporary Technology Fusion: The Case of Intelligent Robots," *Asian J. Technol. Innov.* 15 (2), 2007, pp. 45-65. Available: <http://dx.doi.org/10.1080/19761597.2007.9668637>
- [22] H. A. Grebla, C. O. Cenan, and L. Stanca. *Knowledge Fusion in Academic Networks*. BRAIN: Broad Res. Artif. Intell. Neurosci. 1 (2), Available: <http://www.edusoft.ro/brain/index.php/brain/article/download/60/145>, 2010.
- [23] M. J. Roemer, G. J. Kacprzynski, and R. F. Orsagh, "Assessment of Data and Knowledge Fusion Strategies for Prognostics and Health Management," in *2001 IEEE Aerospace Conference*, vol. 6, 2001, pp. 2979-2988. <http://dx.doi.org/10.1109/AERO.2001.931318>
- [24] K. B. Laskey, P. Costa, and T. Janssen, "Probabilistic Ontologies for Knowledge Fusion," in *11th IEEE International Conference on Information Fusion*, 2008.
- [25] C. Jonquet, P. LePendu, S. Falconer, A. Coulet, N. F. Noy, M. A. Musen, and N. H. Shah, "NCBO Resource Index: Ontology-Based Search and Mining of Biomedical Resources," *J. Web Semant.* 9 (3), 2011, pp. 316-324. Available: <http://dx.doi.org/10.1016/j.websem.2011.06.005>
- [26] L. Y. Lin and Y. J. Lo, "Knowledge Creation and Cooperation between Cross-Nation R&D Institutes," *Int. J. Electron. Bus. Manag.* 8 (1), 2010, pp. 9-19.

- [27] A. Smirnov, M. Pashkin, N. Chilov, T. Levashova, and F. Haritatos, "Knowledge Source Network Configuration Approach to Knowledge Logistics," *Int. J. Gen. Syst.* 32 (3), 2003, pp. 251–269. Available: <http://dx.doi.org/10.1080/0308107031000107766>