AGENT-BASED INTELLIGENT SUPPORT TO COALITION OPERATIONS: A CASE STUDY OF HEALTH SERVICE LOGISTICS SUPPORT

Alexander SMIRNOV, Michael PASHKIN, Nikolai CHILOV, Tatiana LEVASHOVA, and Andrew KRIZHANOVSKY

Abstract: Coalition operations are very likely based on a number of different groups of people, non-governmental organizations, institutions providing humanitarian aid and also army troops and official governmental initiatives. As a result, to manage any coalition operation an efficient knowledge sharing between multiple participating parties is required. The paper proposes an approach that combines knowledge logistics and information fusion at level two of situation assessment. The approach is based on such advanced information technologies as intelligent agents, ontology management, and constraint satisfaction/propagation. The main aim of the paper is to present the developed multi-agent architecture for intelligent support of coalition operations that would enable efficient real-time coalition operation management on-the-fly. A particular attention is given to the following three novel tasks: (i) ontology-driven knowledge representation via object-oriented constraint networks, (ii) processing of free text requests, and (iii) design of adaptive agents for on-the-fly problem solving. As an example of a coalition operation, a fictitious but illustrative case study of mobile hospital configuration from the area of health service logistics is described.

Keywords: Multi-agent Systems, Knowledge Logistics, Ontology Management, Constraint Satisfaction/ Propagation, Coalition Operations, Humanitarian Operations, Free-Text Processing.

Introduction

Coalition operations are very likely based on a number of different, quasi-volunteered, vaguely organized groups of people, non-government organizations, institutions providing humanitarian aid and also army troops and official governmental initiatives. Hence, participants in a coalition operation have to share information within some specified community.[1] As a result, to manage any coalition operation an efficient knowledge sharing between multiple participating parties is required. This knowledge must be pertinent, clear, and correct, and it must be timely processed and delivered to appropriate locations, so that it could provide for situation awareness. This is even more important when an operation involves coalitions uniting resources both of governmental (military, security service, community service, etc) and non-governmental organizations.

Thereby, systems aimed at intelligent support of coalition operations have to meet a number of requirements including (i) support of knowledge sharing, (ii) distributed architecture for collaborative work, (iii) interoperability with other information systems, (iv) dynamic (on-the-fly) problem solving, (v) ability to work with uncertain information, (vi) constraint satisfaction notation for real-world problem description, and other.

Since successful coalition operation can be achieved through knowledge of the status and the dynamics of the situation elements and comprehension of the situation, it can be stated that the right knowledge from distributed sources has to be integrated and transferred to the right person within the right context at the right time to the right purpose. The aggregate of these interrelated activities is referred to as *Knowledge Logistics*.[2]

Knowledge logistics (KL) takes place in a network-centric environment. Unlike hierarchical organizations with fixed commander-subordinate relationships, nodes of network-centric environment are autonomous decision making units that can serve other units and also be served by them. With regard to computer systems, the network-centric environment is based on advanced information technologies such as intelligent agents, ontology management, Web intelligence, Semantic Web, and markup languages. Coalition operation support in the network-centric environment requires rapid processing and analysis of large body of up-to-date (preferably real-time) information from distributed and heterogeneous sources (experts, electronic documents, real-time sensors, weather forecasts, etc.). Hence, one of the key components of the situational awareness is fusion of information from different sources. The most influential fusion model in the area of information fusion is JDL Data Fusion Model.[3] It combines five levels of fusion: 0) sub-object data assessment, 1) object assessment, 2) situation assessment, 3) impact assessment, and 4) process refinement.

The approach proposed in this article combines KL and information fusion at level two of situation assessment and is based on advanced information technologies such as intelligent agents, ontology management, and markup languages. The aim of the paper is to present the developed multi-agent architecture for intelligent support to coalition operations that would enable efficient real-time coalition operation management on-the-fly. The architecture is presented via a case study of an on-the-fly portable hospital configuration problem as one of the major health service logistics support problems. In the proposed approach, the problem is described by a request that is processed by the system implementing the approach. The scenario is the following:

- Entering of the request into the system.
- Association of the request with an application ontology describing the problem:
    - application ontology creation;
    - translation of the request into system's notation and terminology.
- Knowledge processing for answer generation and presentation of results to the user
    - finding appropriate knowledge sources;
    - extraction and translation of knowledge from the sources;
    - fusion of the knowledge and generation of the answer.

All stages of request processing are described in detail in this article.

KSNet-Approach

The KL problem in the presented approach is considered as configuration of a network including end-users, knowledge resources, and a set of tools and methods for knowledge processing located in the network-centric environment. Such network of loosely coupled sources will be referred to as knowledge source network or KSNet,[4,5] and the approach is called KSNet-approach. Correspondingly, the system built as a prototype of the approach is referred to as KSNet-system.

The approach is built upon constraint satisfaction/ propagation technology for problem solving since application of constraint networks allows simplifying the formulation and the interpretation of real-world problems that are usually represented as constraint satisfaction problems in such areas as management, engineering, etc.[6]. ILOG [7] has been selected as a constraint satisfaction/propagation.

As it has already been mentioned, the KSNet-approach addresses both KL and information fusion based on constraint satisfaction methodology. In accordance with the Endsley's situation awareness model,[8] the approach currently covers two first levels: perception of elements in the current situation and comprehension of the current situation. The third level of future status projection is planned as future work.

The multi-agent architecture, based on FIPA's Reference Model [9] as a technological basis for definition of agents' properties and functions, is presented in Figure 1. The FIPA-based technological kernel agents used in the KSNet-system are: *wrapper*, *facilitator*, *mediator*, and *user agent*. The designed problem-oriented agents and their

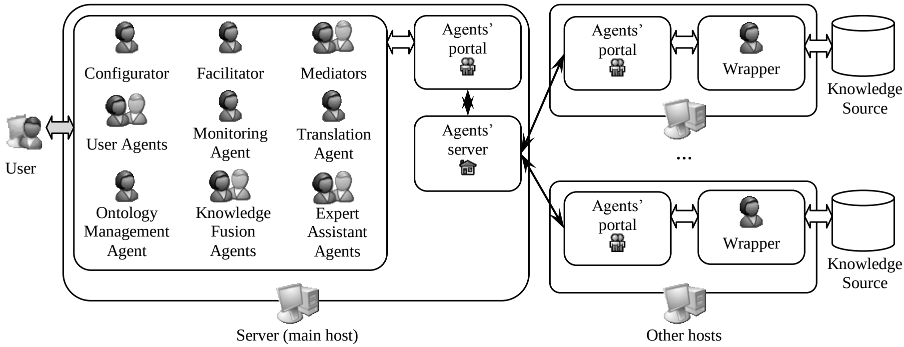

Figure 1: The Multi-Agent Architecture of the KSNet-System.

main tasks are: *translation agent* (performs translation between different vocabularies), *KF agent* (performs operation for KF), *configuration agent* (efficient use of KSNet), *ontology management agent* (performs ontology operations), *expert assistant agent* (interaction with experts), and *monitoring agent* (knowledge sources verification). The agents communicate via a predefined scenario (see Figure 2). The most interesting, new and complex agents are described in detail in the paper; the rest are only briefly mentioned.

Portable Hospital Configuration: The Request

The nature of coalition-based operations has spanned a broad range of missions from war to operations other than war (OOTW). The OOTW cover a range of missions where sides are not in direct conflict but are required to perform a "neutral third party" operation. This is usually the result of a situation that is beyond the capability of the individual sides to resolve because it is an internecine issue or is beyond their individual resources. The missions may be further subdivided into war avoidance and humanitarian aid missions. The war avoidance operations cover the spectrum of "policing" activities that are required to restore "peaceful normality" in hostile situations between two or more population elements in conflict. In these circumstances, the allied forces must act as an independent arbiter or "referee." Countering terrorism and international crime may also be considered to lie within such missions because they can also have a significantly destabilizing influence and may require the co-operation of international agencies in order to limit their insidious effects.

With the above mentioned factors in mind, "Binni – Gateway to the Golden Bowl of Africa" is a hypothetical scenario. Binni is in the Sudanese Plain. The countries of Gao, Agadez and Binni are fictitious, as are the events, organizations and personalities that lead to the crisis requiring UN intervention. However, it provides a backdrop

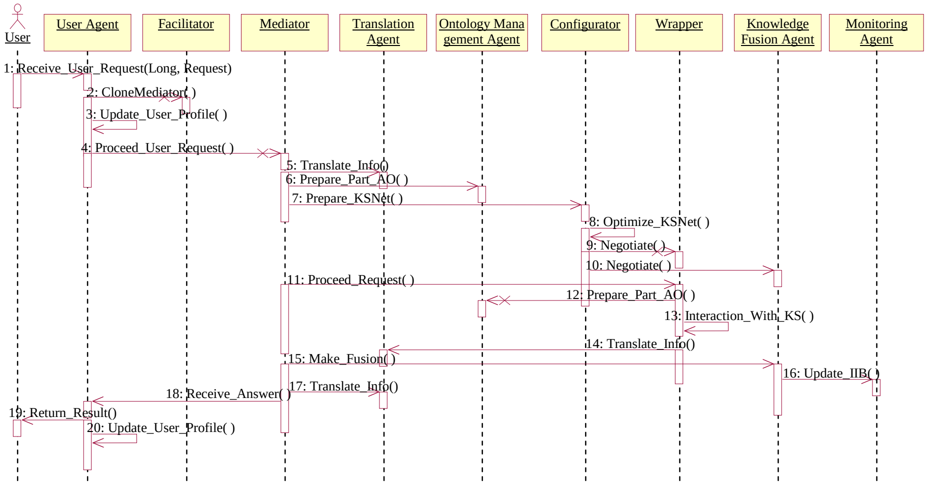

Figure 2: Agent Communication Scenario during User Request Processing in the KSNet-System

against which to develop a number of exercises typical of those anticipated for future coalition force operations. Details of the scenario are given in a comprehensive document developed for the DARPA CoABS (Control of Agent Based Systems) program.[10] This case study was used in other works related to multi-agent environments.

The Request

In the presented case study a hospital construction task is considered. Configuration of a mobile or portable hospital is an actual topic in such areas as counter-terrorism.[11] Such a mobile hospital can be characterized as a portable (maybe even disposable) structure, which can be built in a limited amount of time with given characteristics such as number of beds, location, etc. In the presented case study, construction of such a hospital in the Binni region is considered.

Experimenting with the Binni scenario intends to demonstrate how the developed KSNet-approach can be used for support of coalition-based OOTW.

The following request is considered:

> *Define suppliers, transportation routes and schedules to build a hospital of given capacity at given location in given time.*

To process the request, three major tasks were selected: (i) request input, (ii) request recognition, and (iii) request processing tracking. The following agents were assigned to these tasks: user agent, translation agent, and monitoring agent.

Facilitator

Facilitator is a FIPA technological agent. It provides “yellow pages” service for other agents in multi-agent systems. This makes it possible for the agents to find partners for conducting certain tasks.

User Agent

User agent is a FIPA technological agent responsible for providing an interface with the user, and customizes his/ her support during interaction with the system. For the considered case of free-text request this agent does not perform any critical functions and is not considered in detail.

Translation Agent

When a request enters the system for the first time it cannot be recognized automatically and, therefore, this task is performed by experts. Later, when application ontology is built (see the next section), this can be done automatically by the translation agent. The procedure is described further in this article.

Mediator

Mediator is a FIPA technological agent responsible for tracking all stages of request processing. It finds appropriate free agents and assigns to them different tasks (e.g., “recognize the request” for translation agent).

Application Ontology Creation

Due to the fact that KL assumes dealing with knowledge contained in distributed and heterogeneous sources, the approach is oriented towards an ontological model providing for a common way of knowledge representation that supports semantic interoperability. A fundamental ontology providing a common notation implemented through an ontology library lies at the core of the framework. The ontology library is a central knowledge repository that prescribes a common notation and provides a common vocabulary for the ontologies that it stores. The common representation tools/ aids enable performance of operations on ontology integration such as alignment and merging, and operations on context integration.

Main components of the ontology library are domain, tasks & methods, and application ontologies. All these ontologies are interrelated so that application ontology (AO) is a specialization both of domain and tasks & methods ontologies. Tasks in tasks & methods ontology represent types (classes) of formalized problems. Since ontologies of different domains are stored in the ontology library (every domain is represented by its domain ontology), AO can specialize in the knowledge of several domains. Therefore, AO plays two roles: it serves as cross-domain ontology and represents the context of the problem to be solved.

Domain ontologies and tasks & methods ontologies are formed as new knowledge becomes available. The new knowledge here is the knowledge provided by experts, retrieved from knowledge sources (KSs), or obtained as a result of users' requests processing. Both new ontologies can be created (if there is no ontology related to the domain/ task/ method of the new knowledge) and existing ontologies can be expanded (otherwise). Such an arrangement makes it possible to continuously enrich the ontology library with new knowledge, and, consequently, we may talk about system's learning capability.

Being a context-dependent conceptual model that describes a real-world application domain depending on a specific user request and relevant to its particular domains and tasks, AO plays a central role in request processing; it also represents the common knowledge of the user and KSs. AO is formed by merging the request relevant parts of domain and tasks & methods ontologies into a single ontology. The information requested from KSs is associated with AO formed for the processing of the request. Request ontologies and KS ontologies are used to translate the users' and KS terms into AO terms.

In accordance with the ontology spectrum presented in the work of McGuiness [12] the ontologies used in the KSNet-approach correspond to the level of value restrictions and tasks & methods ontologies can be related to the level of general logic.

As a general model of ontology representation in the KSNet-system, an object-oriented constraint network paradigm was used (see Figure 3). This model defines the common ontology notation used in the system. According to this representation an ontology (A) is defined as $A = (O, Q, D, C)$, where O is a set of *object classes* ("*classes*"). Each of the entities in a class is considered as an *instance* of the class. Q is a set of class attributes ("*attributes*"). D is a set of attribute domains ("*domains*"). And C is a set of *constraints*.

For the chosen notation the following six types of constraints have been defined $C = C^I \cup C^{II} \cup C^{III} \cup C^{IV} \cup C^V \cup C^{VI}$: $C^I = \{c^I\}$, $c^I = (o, q)$, $o \in O$, $q \in Q$ – accessory of attributes to classes; $C^{II} = \{c^{II}\}$, $c^{II} = (o, q, d)$, $o \in O$, $q \in Q$, $d \in D$ – accessory of domains to attributes; $C^{III} = \{c^{III}\}$, $c^{III} = (\{o\}, True \vee False)$, $|\{o\}| \geq 2$, $o \in O$ – classes compatibility (compatibility structural constraints); $C^{IV} = \{c^{IV}\}$, $c^{IV} = \langle o', o'', type \rangle$, $o' \in O$, $o'' \in O$, $o' \neq o''$ – hierarchical relationships (hierarchical structural constraints) "is a" defining class taxonomy ($type$=0), and "has part"/"part of" defining class hierarchy ($type$=1);$C^V = \{c^V\}$, $c^V = (\{o\})$, $|\{o\}| \geq 2$, $o \in O$ – associative relationships ("one-level" structural constraints); $C^{VI} = \{c^{VI}\}$, $c^{VI} = f(\{o\}, \{o, q\}) = True \vee False$,

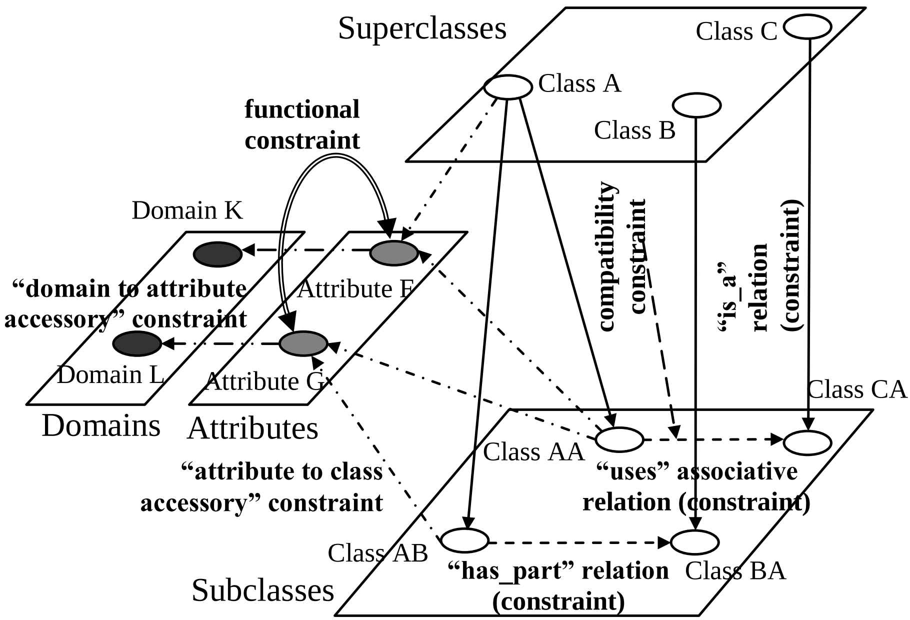

Figure 3: Object-Oriented Constraint Network Paradigm.

$|\{o\}| \geq 0$, $|\{q\}| \geq 0$, $o \in O$, $q \in Q$ – functional constraints referring to the names of classes and attributes.

- In order to build AO for this case study, parts of ontologies corresponding to the described task were found in Internet's ontology libraries (see Table 1). The analyzed ontologies represent a hospital in different manners. The resulting ontology is shown in Figure 5.
- *portable hospital* is a class ($o_1 \in O$);
- *material costs* is an attribute ($q_1 \in Q$);
- domain 1 ($d_1 \in D$) consists of *positive real numbers* (R^+);
- the attribute *material costs* (q_1) belongs to the class *portable hospital* (o_1): $c^I_1 = (o_1, q_1)$;
- the attribute *material costs* (q_1) belonging to the class *portable hospital* (o_1) may take positive real values: $c^{II}_1 = (o_1, q_1, d_1)$;
- a class *hospital bed* (o_2) is compatible with a class *hospital bed supplier* (o_3): $c^{III}_1 = (\{o_2, o_3\}, True)$.

Table 1: Ontologies used for Building "Hospital Configuration" Application Ontology.

Ontology	URL	Format
Clin-Act (Clinical Activity), the library of ontologies [13]	http://saussure.irmkant.-rm.cnr.it/onto/	KIF
Upper Cyc/HPKB IKB ontology with links to SENSUS, Version 1.4 [14]	http://www-ksl-svc.stanford.-edu:5915	Ontolingua (KIF)
Loom ontology browser, Information Sciences Institute, The University of Southern California [15]	http://sevak.isi.edu:4676/loom/-shuttle.html	Loom
North American Industry Classification System (NAICS) code, DAML Ontology Library [16]	http://opencyc.sourceforge.net/-daml/naics.daml	DAML
The UNSPSC Code (Universal Standard Products and Services Classification Code), DAML Ontology Library, Stanford University [17]	http://www.ksl.stanford.edu/-projects/DAML/UNSPSC.daml	DAML
Web-Onto [18]	http://eldora.open.ac.uk:-3000/webonto	OCML

In the given case study the following examples of ontology elements can be presented:

- an instance of the class *furniture* (o_4) can be a part of an instance of the class *portable hospital* (o_1): $c^{IV}{}_1 = \langle o_1, o_4, 1\rangle$;
- a *hospital bed* (o_2) is *furniture* (o_4): $c^{IV}{}_2 = \langle o_2, o_4, 0\rangle$;
- an instance of the class *hospital bed* (o_2) can be connected to an instance of the class *hospital bed supplier* (o_3): $c^{V}{}_1 = (o_2, o_3)$;
- the value of the attribute *material cost* (q_1) of an instance of the class *portable hospital* (o_1) depends on the values of the attribute *cost* (q_2) of instances of the class *furniture* (o_4) connected to that instance of the class *facility* and on the number of such instances: $c^{VI}{}_1 = f(\{o_4\}, \{(o_1, q_1), (o_4, q_2)\})$.

These examples do not pretend to be complete definition of the ontology due to the fact that the entire AO is very complex.

When dealing with knowledge, uncertainties may arise for the following reasons: (i) lack of information, (ii) invalidity of information, (iii) subjectivity, (iv) lack of knowledge about a problem, (v) unverbalizability of the problem, (vi) imprecision of the problem-solving methods. This is especially important for disaster response operations because such operations have to be fast and they take place in conditions of damaged infrastructure, including communications. For the purpose of processing uncertain knowledge, fuzzy constraint satisfaction is used. The object-oriented constraint network with uncertainties is represented below.

In accordance with the formalism, a fuzzy object-oriented constraint network is described as $(O, Q, D, C_\mu, W, T, I_p)$, where C_μ is a set of constraints (each constraint contains a function of membership μ to [0, 1] associated to weight ω_c representing its weight (importance) or priority); W is a weighting scheme, i.e. a function combining the degree of satisfaction of constraint $\mu(c)$ to ω_c for estimation of the weighted degree of satisfaction of $\mu^\omega(c)$; T is an aggregation function, which performs simple partial regulation on defined values, defining C_μ; and I_p is an information content (objects – instances of classes) of the constraint network, which is of a nondeterministic or probabilistic nature.

Constraints of belonging of attributes to classes, compatibility structural constraints, hierarchical structural constraints and “one-level” structural constraints are hard constraints. All of them have to be satisfied in the solution, i.e. for each of them $\omega_c = 1$. Functional constraints and accessory of domains to attributes can be considered as soft constraints. An ontology engineer assigns a degree of necessity to functional constraints. If the KF agent cannot find a feasible solution for a user request it can omit constraints with low degree of necessity.

Constraints of accessory of domains to attributes were modified in the following way. For interval domains $d = [d_1, d_2]$ a degree of belief was introduced (see Figure 4). On this interval the degree of belief to attribute value P is equal to “1.” For this interval an expert can assign Δ_1 and Δ_2 – deviations of attribute value from interval boundaries. By means of these deviations it is possible to expand the number of instances of classes from KSs and to estimate the quality of system’s response. For example, the number of operating tables per 50 patients (x) can be described as:

$$\omega_c(x) = \begin{cases} 0, x \le 1; x \ge 8 \\ \dfrac{x-1}{3}, 1 < x < 4 \\ 1, 4 \le x \le 8 \\ \dfrac{8-x}{2}, 6 < x < 8 \end{cases}$$, where $\omega_c(x)$ is the fuzzy value of the constraint c; and x

is the number of operating tables per 50 patients.

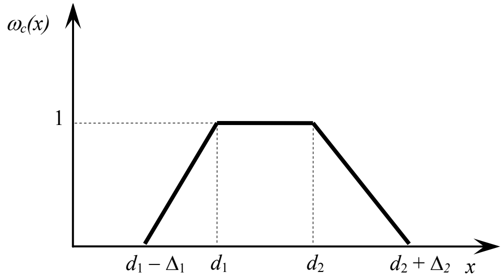

Figure 4: An Example of Degree of Belief for Interval Domain.

Ontology Management Agent

The ontology management agent is responsible for operations with ontologies and provides services for other agents when they need to access certain ontologies or parts of ontologies.

Translation Agent

As it has been mentioned above, there are two cases requiring request recognition: (i) when a request enters the system for the first time and (ii) when similar requests have already been processed and AO for such problem already exists. In this article the second case is considered.

While analyzing text recognition techniques, it has been found that there are no algorithms that could provide reliable results for context-independent texts. Therefore, the authors decided to develop a new algorithm based on the application of ontology.

This algorithm could not be considered as a result of complex linguistic research; however, it appeared to be efficient enough besides the fact that it has a number of limitations such as language dependence and a limited number of syntax constructions to be recognized. The algorithm consists of the following operations:

- *Tokenization*. Identification of borders of words, numbers, etc.
- *Stop-words finding*. The translation agent has a list of stop words that should be omitted during request recognition (e.g., articles, some prepositions, etc.).
- *Spelling*. This helps to identify misspelled words that can be later matched with ontology elements.
- *Stemming*. This operation makes it possible to identify ontology elements even if they are written in different forms (e.g., patients → patient).
- *Synonyms*. At this stage the request is extended with synonyms for all words. This allows for example to identify class “*hospital*” if the user enters “*clinic*.”

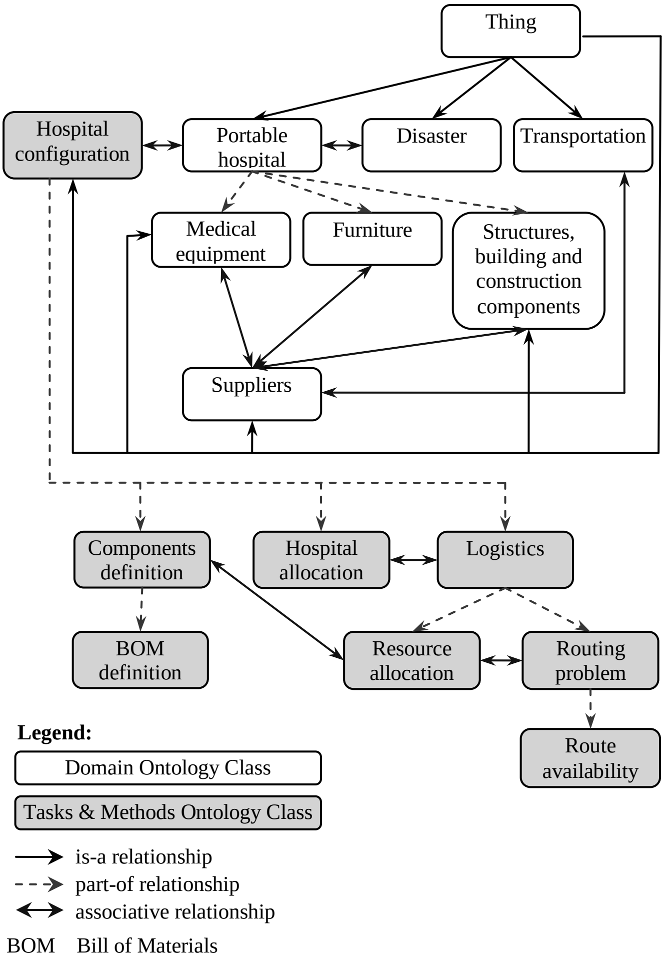

Figure 5: AO for Hospital Configuration Problem.

- *Ontology search*. Here all the words found so far are matched with ontology elements. As a result the following groups of words are identified and recorded into an XML-file (see Figure 6):
    - *Class*, if the word is located after a definite or indefinite article ("*the*" or "*a*") and before a preposition (e.g., "*with*") or subordinating conjunction (e.g., "*where*").
    - *Task*, if the word is located after the particle "*to*" and before a definite or indefinite article, e.g. "*to build a hospital*."

```
<?xml version="1.0" standalone="yes"?>
<!--KSNet 2002 SPIIRAS-->
<Root_Element>                       <!--This is an xml comment         -->
    <KSWord><!--Beginning of the information block for the word «build»-->
        <Word>build</Word>           <!--Keyword is «build»             -->
        <Type>Action</Type>          <!--Type of word: «build»          -->
        <Synonym>construct</Synonym> <!--Synonym of «build»             -->
        <Synonym>make</Synonym>      <!--Synonym of «build»             -->
        <Synonym>create</Synonym>    <!--Synonym of «build»             -->
    </KSWord>      <!--End of the information block for word «build»    -->
    <KSWord>       <!--Phrase «mobile hospital» has no synonyms         -->
        <Word>mobile hospital</Word>
        <Type>Class</Type>       <!--Type of phrase: «mobile hospital» -->
    </KSWord>
    <KSWord>
        <Word>price</Word>           <!--Keyword «price»                -->
        <Type>Attribute</Type>       <!--Type of «price <= 1000»        -->
        <Operator>&lt;=</Operator>   <!--Type of constraint «<=»        -->
        <Value>1000</Value>          <!--Value of constraint «1000»     -->
        <Synonym>terms</Synonym>     <!--Synonym of «price»             -->
        <Synonym>cost</Synonym>      <!--Synonym of «price»             -->
    </KSWord>
</Root_Element>
```

Figure 6: Example Results for Request Recognition.

  - *Attribute* contains three components: (i) name of attribute (tag <Word>), (ii) type of constraint (tag <Operator>), and (iii) value of constraint (tag <Value>) (the different types of constraints are: <, >, =, ≤, ≥, ≠).
  - *Unrecognized word*, if the word is absent in the ontology and cannot be related to any of the above described types.

  Combination of matching ontology elements is referred to as "*structural constituent*" and attribute values are referred to as "*parametric constituent*."

- *Normalization*. At this stage the elements of the parametric constituent are recalculated. For instance, if distance in the request is given in kilometers but in the application ontology it is measured in miles, the value will be changed accordingly.

When the recognition of the request is finished, the results are passed to the *Mediator* and knowledge processing for that request begins.

Knowledge Processing

The main idea of this stage is to use the application ontology as a constraint satisfaction task, to define the goal and some parameters from the request, and to "fill" the rest of the parameters with knowledge from different knowledge sources.

Wrapper

The system could work with various knowledge sources: databases, knowledge bases, structured documents, and experts. This list is not complete and can be extended. Due to the fact that each knowledge source or several similar knowledge sources are of different nature, they have to be processed in different ways. For this purpose wrapper agents are used. These agents "sit" on knowledge sources and "know" their structure and notation. On the other side, wrappers communicate with the system and support its notation of the object-oriented constraint network and terminology of the application ontology. The translation is done using knowledge source ontologies that set correspondence between elements of knowledge sources and those of application ontology, and define functions to convert certain values (e.g., kilometers into miles).

Expert Assistant Agent

The expert assistant agent performs the same functions as the wrapper but it is intended to communicate with the experts due to the fact that the experts need a convenient and user-oriented interface. Thereby it combines functions of wrapper and user agents.

Configuration Agent

Processing of user requests assumes acquisition of knowledge from KSs. Since there are many knowledge sources, it is necessary to define which are to be used in each particular case. Besides, some of the sources may contain the same information, i.e. to be alternative. As a result the task of choosing appropriate sources appears. The goal of this task is selection of KSs to be used for user request processing in a most efficient way according to predefined criteria such as costs, time, and reliability. For the given request, knowledge sources containing the following information were considered:

- Hospital related information (constraints on its structure, required quantities of components, required times of delivery);
- Available friendly suppliers (constraints on suppliers' capabilities, capacities, locations);
- Available friendly providers of transportation services (constraints on available types, routes, and time of delivery);
- Geography and weather of the Binni region (constraints on types, routes, and time of delivery, e.g. by air, by trucks, by off-road vehicles);
- Political situation, e.g. who occupies the territory used for transportation, existence of military actions on the routes, etc. (additional constraints on routes of delivery).

The configuration agent implements an engine that solves this task. For the task a number of solution techniques were tried and a Genetic Algorithm (GA) was chosen as a probabilistic approach to pseudo-optimal solutions search. It is well suited for tasks of enumeration nature. From the other techniques, k-nearest neighbor (this technique was computationally intensive for large data sets) and decision tree techniques (this technique did not provide satisfactory reduction of the decision space) are worth mentioning.

To investigate the efficiency of GA for the task of choosing KSs, a set of experiments with a basic GA for tasks of different dimensions were performed, with KSs' parameters and knowledge maps being randomly generated. The results indicate that GA applied to the problem of choosing KSs behaves as expected: the number of required calculations for obtaining a quasi-efficient decision using even basic non-optimized GA is smaller than that in the method of exhaustive search. Figure 7 represents the ratio of calculations number for the method of exhaustive search to that for the GA. As the task dimension grows this improvement grows nonlinearly. This proves that the application of GA to the problem of choosing KSs is justified.

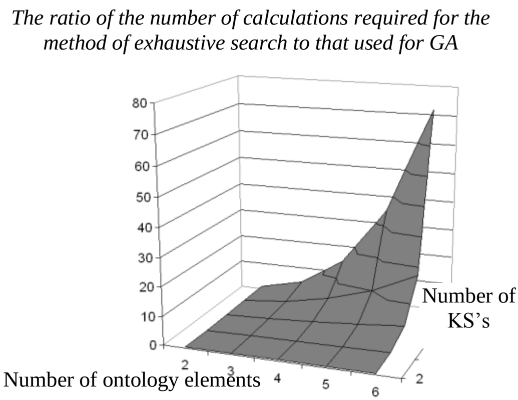

Figure 7: Efficiency Improvement due to GA Application.

Monitoring Agent

Since the knowledge sources, especially in disaster response and evacuation operations, are dynamic, continuous run-time monitoring and tracking is one of the key factors for success. In the presented system a monitoring agent is provided for permanent checking of the knowledge sources for updated information about the current situation. It enables timely planning of the activities in a network-centric environment since decision makers will have updated information.

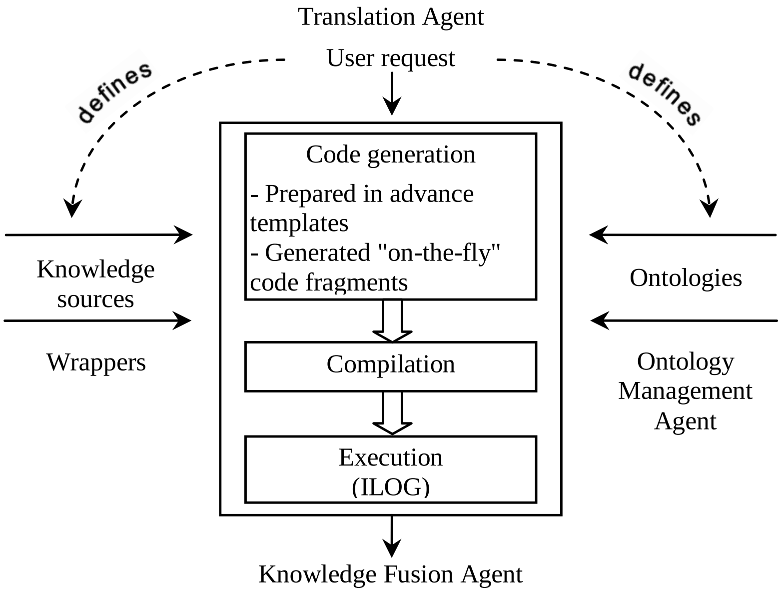

Figure 8: The Concept of the on-the-fly Compilation Mechanism.

Knowledge Fusion Agent

The knowledge fusion agent performs fusion of structured knowledge from the application ontology and structural constituent of the request and instantiated knowledge from knowledge sources and parametric constituent. Due to the chosen formalism of object-oriented constraint networks, the fusion task can be solved by composing and solving a constraint satisfaction problem. As it has been mentioned above, ILOG was selected as a constraint satisfaction/ propagation technology for the approach, considering the fact that it incorporates a powerful constraint solver that provides mechanisms for easy definition of the problem to be solved and constraint propagation mechanisms. It is based on C++ libraries and thereby it can be easily integrated into various programs and applications.

Although the task of defining a constraint satisfaction problem is the same for different problems, the task of solving it differs from a request to a request. For this purpose the described approach implements adaptive knowledge fusion agents. These agents could modify themselves while solving a particular task. Upon receiving a task the agents load an appropriate ontology and generate an executable module for solving it "on-the-fly."

In the proposed approach a novel "on-the-fly" compilation mechanism is proposed to solve varying problems. In a rough outline this novel "on-the-fly" compilation mechanism is based on the following concepts (see Figure 8):

- A pre-processed user request defines (1) which ontologies are to be used for the problem domain description, and (2) which knowledge sources are to be used;
- C++ code is generated on the basis of information extracted from (1) the user request (goal, goal objects, etc.), (2) appropriate ontologies (classes, attributes, and constraints), and (3) suitable knowledge sources;
- Compilation is performed in an environment of prepared in advance C++ project;
- Failed compilations/ executions do not fail the work of the whole system; an appropriate error message is generated.

The essence of the proposed on-the-fly compilation mechanism is to write the ontology elements (classes, attributes, constraints) to a C++ file directly, so that it could be compiled into an ILOG-powered program. The service responsible for problem solving creates the C++ file based on these data and inserts the program source code into the prepared in advance Microsoft Visual Studio project. The program is compiled and as a result an executable file in the form of dynamic-link library (DLL) is created. After that, the service calls a function from the DLL to solve the problem. The performed experiments demonstrated that for complex tasks the compilation time is significantly less than the time required for task solving by the generated program.

Results

The results are presented to the user by the user agent. Depending on the request parameters and the answer generation procedure the results can be delivered on-line (as a web-page), or off-line (as an e-mail message or sms). The example results for defining routes for the task of hospital components delivery are presented in Figure 9. The figure uses the following notation. Small dots are the cities of the region. The city indicated with a pin (Aida) is the closest city to the disaster (indicated with a cross), where the mobile hospital is to be built. The bigger dots are the cities where suppliers are situated and they have to be visited (Libar, Higgville, Ugwulu, Langford, Nedalla, Laki, and Dado). Transportation routes are shown as lines. The lines with trucks denote the routes of particular vehicle groups (indicated with appropriate callouts). Other lines are routes that are not used for transportation in the solution. The lines attached to closed cities (indicated with thunderstorms next to them) are not available for transportation.

Conclusions

This article presented an approach that implements KL for intelligent support of OOTW. Within the approach, KL is coupled with information fusion based on constraint satisfaction methodology. Applying constraint networks allows rapid problem manipulation by adding/ changing/ removing its components (objects, constraints, etc.) and use of existing efficient constraint satisfaction/ propagation technologies such as ILOG.

The presented mechanism of on-the-fly problem solving makes it possible to solve dynamic constraint satisfaction problems as a sequence of static ones. The agent-based architecture increases scalability, efficiency and interoperability of the KSNet-system. Utilizing ontologies and the compatibility of the employed ontology notation with modern standards (such as DAML+OIL) enables semantic interoperability with other knowledge-based systems and services and facilitates knowledge sharing.

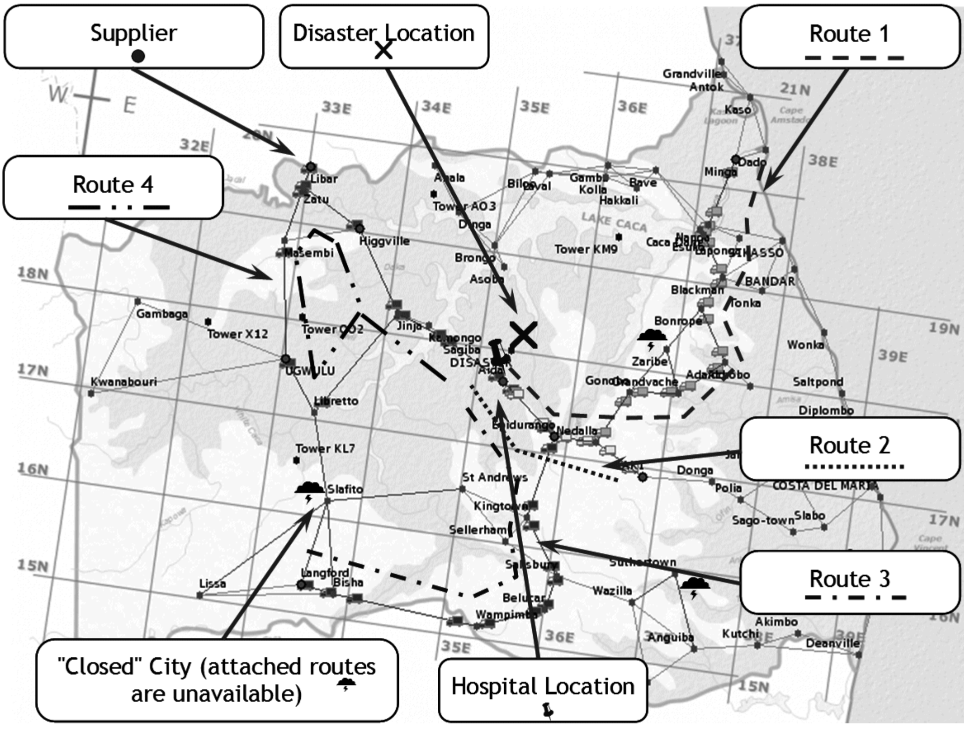

Figure 9: Example Routing Plan.

Acknowledgements

Some parts of the research were done as parts of the ISTC partner project #1993P funded by Air Force Research Laboratory at Rome, NY; project #16.2.44 of the research program “Mathematical Modelling and Intelligent Systems;” project #1.9 of the research program “Fundamental Basics of Information Technologies and Computer Systems” of the Russian Academy of Sciences; and grant #02-01-00284 of the Russian Foundation for Basic Research. Some prototypes were developed using software granted by ILOG Inc.

Notes:

[1] Michal Pechoucek, Vladimír Marík, and Jaroslav Barta, “CplanT: an Acquaintance Model-Based Coalition Formation Multi-Agent System” (paper presented at the Second International Workshop of Central and Eastern Europe on Multi-Agent Systems (CEEMAS’2001), Krakow, Poland, September 2001), 234-241.

[2] Alexander Smirnov, Mikhail Pashkin, Nikolai Chilov, Tatiana Levashova, and F. Haritatos, “Knowledge Source Network Configuration Approach to Knowledge Logistics,” *International Journal of General Systems* 32, no. 3 (2003): 251-269.

[3] John Salerno, Michael Hinman, Doug Boulware, and Paul Bello, “Information Fusion for Situational Awareness” (paper presented at the 6th International Conference on Information Fusion, Cairns, Queensland, Australia, July 2003), 507-513.

[4] Alexander Smirnov, Mikhail Pashkin, Nikolai Chilov, and Tatiana Levashova, “Knowledge Logistics in Information Grid Environment,” *Future Generation Computer Systems* 20, no. 1 (January 2004), special issue “Semantic Grid and Knowledge Grid: The Next-Generation Web,” ed. H. Zhuge: 61-79.

[5] Alexander Smirnov, Mikhail Pashkin, Nikolai Chilov, and Tatiana Levashova, “Agent-Based Support of Mass Customization for Corporate Knowledge Management,” *Engineering Applications of Artificial Intelligence* 16, no 4 (2003): 349-364.

[6] Hartwig Baumgaertel, “Distributed Constraint Processing for Production Logistics,” *IEEE Intelligent Systems* 15, no 1 (2000): 40-48.

[7] ILOG corporate web-site <http://www.ilog.com> (16 July 2004).

[8] Mica R. Endsley, “Theoretical Underpinnings of Situation Awareness: A Critical Review,” in *Situation Awareness Analysis and Measurement*, ed. Mica R. Endsley and Daniel J. Garland (Mahwah, NJ: Lawrence Erlbaum Associates, 2000), 3-32.

[9] Foundation for Intelligent Physical Agents (FIPA) <http://www.fipa.org> (16 Jul 2004).

[10] R.A. Rathmell, “A Coalition Force Scenario ‘Binni – Gateway to the Golden Bowl of Africa’” (paper presented at the International Workshop on Knowledge-Based Planning for Coalition Forces, ed. Austin Tate, Edinburgh, Scotland, 10-11 May 1999), 115-125.

[11] The interactive edition of the Pittsburg Post-Gazette <www.post-gazette.com> (16 July 2004).

[12] Deborah L. McGuiness, “Ontologies for Information Fusion” (paper presented at the 6th International Conference on Information Fusion, Cairns, Queensland, Australia, July 2003), 650-657.

[13] Clin-Act (Clinical Activity), The ON9.3 Library of Ontologies: Ontology Group of IP-CNR (a part of the Institute of Psychology of the Italian National Research Council (CNR)) <http://saussure.irmkant.rm.cnr.it/onto/> (30 Dec 2002).

[14] Hpkb-Upper-Level-Kernel-Latest: Upper Cyc/HPKB IKB Ontology with links to SENSUS, Version 1.4, 1998. Ontolingua Ontology Server <http://www-ksl-svc.stanford.edu:5915> (30 Dec 2002).

[15] Loom ontology browser, Information Sciences Institute, The University of Southern California <http://sevak.isi.edu:4676/loom/shuttle.html> (30 Sep 2002).

[16] North American Industry Classification System code, DAML Ontology Library, Stanford University <http://opencyc.sourceforge.net/daml/naics.daml> (30 Dec 2002).

[17] The UNSPSC Code (Universal Standard Products and Services Classification Code), DAML Ontology Library, Stanford University <http://www.ksl.stanford.edu/projects/DAML/UNSPSC.daml> (30 Dec 2002).

[18] WebOnto: Knowledge Media Institute (KMI), The Open University, UK <http://eldora.open.ac.uk:3000/webonto> (30 Dec 2002).

ALEXANDER V. SMIRNOV, Prof., received his ME, PhD and D.Sc. degrees in St. Petersburg, Russia, in 1979, 1984, and 1994, respectively. He is Deputy Director for Research and Head of Computer Aided Integrated Systems Laboratory at St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS). He is full professor of St. Petersburg State Polytechnic University and St. Petersburg State Electrical Engineering University and member of IEEE. His current research interests belong to areas of corporate knowledge management, multi-agent systems, group decision support systems, virtual enterprises, and supply chain management. He has published more than 200 research papers in reviewed journals and proceedings of international conferences, books, and manuals.

MIKHAIL P. PASHKIN received his MAM from St. Petersburg State University, Russia, in 1992. He is researcher at the Computer Aided Integrated Systems Laboratory of SPIIRAS. His current research interests include multi-agent systems, ontology engineering, and group decision support systems. He has published more than 40 research papers in proceedings of international conferences, books and manuals.

NIKOLAI CHILOV received his ME from St. Petersburg State Technical University, Russia, in 1998. He is researcher at the Computer Aided Integrated Systems Laboratory of SPIIRAS. His current research interests include virtual enterprise configuration, supply chain management, knowledge management, ontology engineering and multi-agent systems. He is an author/ co-author of more than 35 research papers published in proceedings of international conferences and books.

TATIANA LEVASHOVA received her ME degree from St. Petersburg State Electrical Engineering University in 1986. She is lead programmer at the Computer Aided Integrated Systems Laboratory of SPIIRAS. Her current research is devoted to knowledge-related problems such as knowledge representation, knowledge management, and ontology management. She has published more than 15 papers in reviewed journals and proceedings of international conferences.

ANDREW KRIZHANOVSKY received his ME from St. Petersburg State Technical University, Russia, in 2002. He is a researcher at the Computer Aided Integrated Systems Laboratory of SPIIRAS. His current research interests include knowledge management, ontology engineering, multi-agent systems, and computer linguistics.