Computing and Informatics, Vol. 22, 2003, ??–??

KSNET-APPROACH TO KNOWLEDGE FUSION FROM DISTRIBUTED SOURCES

Alexander Smirnov, Mikhail Pashkin,
Nikolai Chilov, Tatyana Levashova

St. Petersburg Institute for Informatics and Automation
of the Russian Academy of Sciences
39, 14th Line, St. Petersburg, 199178, Russia
e-mail: smir@iias.spb.su

Manuscript received 11 November 2002; revised 19 June 2003
Communicated by Pavol Návrat

Abstract. The rapidity of the decision making process is an important factor in different branches of the human life (business, healthcare, industry, military applications etc.). Since responsible persons make decisions using available knowledge, it is important for knowledge management systems to deliver necessary and timely information. Knowledge logistics is a new direction in the knowledge management addressing this. Technology of knowledge fusion, based on the synergistic use of knowledge from multiple distributed sources, is a basis for these activities. The paper presents an overview of a Knowledge Source Network configuration approach (KSNet-approach) to knowledge fusion, multi-agent architecture and research prototype of the KSNet knowledge fusion system based on this approach.

Keywords: Knowledge management, intelligent systems, knowledge representation, multi-agent systems, knowledge fusion

1 INTRODUCTION

Current trends of decision making in a wide range of applications require operating in a global information environment. This leads to an expansion of tools dealing with knowledge storing in the Internet based on intensive use of WWW-technologies and such standards as XML, RDF, DAML+OIL, etc. [1, 2]. Thus it is possible to speak about an evolution of the information environment, incorporating end-users and loosely coupled knowledge sources — KSs — (experts, knowledge bases (KBs), repositories, etc.), from "regular" (with fixed interactions between its elements) to "intelligent" (with flexible configuration of knowledge network in which humans are involved). Growing importance of knowledge emerged due to this evolution results in a need for acquisition, integration, and transfer of the right knowledge from distributed sources in the right context to the right person in the right time for the right business purpose. These activities called Knowledge Logistics (KL) are required for global awareness, dynamic planning and global information exchange in the information environment.

The approach to KL through knowledge fusion (KF) described here, called Knowledge Source Network (KSNet-approach) implies synergistic use of knowledge from different sources in order to complement insufficient knowledge and obtain new knowledge [3]. The architecture developed for the KF system (called KSNet system) is based on this approach and utilizes such technologies as ontology management, intelligent agents, constraint satisfaction, soft computing, and groupware.

Intelligent agents and multi-agent systems are the research topics which significantly changed the distributed systems functioning. Multi-agent systems offer an efficient way to understand, manage and use the distributed, large-scale, dynamic, open, and heterogeneous computing and information systems [4, 5]. In agent-based systems an agent must represent its knowledge in a vocabulary of a specified ontology [6, 7]. Ontologies are a technique of semantic knowledge representation for its further processing, and are considered as content theories of the kinds of objects, properties of objects and relations between objects possible in a specified knowledge domain, i.e. ontologies provide potential terms describing knowledge about the application domain [8]. An object-oriented constraint network paradigm was proposed as a general model of ontology representation in the KF system KSNet based on the KSNet-approach [9].

The paper has the following structure: (i) shortly describes the state-of-the-art of the knowledge management areas, (ii) presents major technologies for KL (KF operations and an ontology-driven methodology), a knowledge repository structure and a multi-agent architecture of the KSNet system, and (iii) describes the research prototype of this system being developed.

2 KNOWLEDGE MANAGEMENT: STATE-OF-THE-ART

Data is a set of facts, which can exist in multiple forms at different locations and often not interconnected. *Information* is a set of facts interconnected and delivered in a clear context and time. Information is more than a sum of its raw elements which can be considered as data. *Knowledge* is a set of relations (constraints, functions, rules) by which a user/an expert decides how, why, where and what to do with the information to timely produce adaptive actions meeting a goal or a set of goals. Knowledge may be considered as a high-value form of information that is ready to be applied to decision and actions.

A research topic dealing with knowledge is Knowledge Management (KM). It is defined as a complex set of relations between people, processes and technology bound together with the cultural standards, like mentoring and knowledge sharing, which constitute an organization's social capital [10]. KM consists of the following tasks: knowledge discovery (knowledge entry, capture of tacit knowledge, KF, etc.), knowledge representation (KB development, knowledge sharing and reuse, knowledge exchange, etc.), knowledge mapping (identifying KSs, indexing knowledge, making knowledge accessible) [11–15]. There are a number of different approaches proposed and tools developed for these tasks solving, based on the algorithms of data searching and retrieving in large databases, technologies of data storing and representation, etc. Among them the following ones can be pointed out: Microsoft SharePoint Portal [16], SearchServer/KnowledgeServer [17], Lotus Discovery Server [18], Text-To-Onto [19], etc. (knowledge searching and retrieving from different types of documents); Disciple-RKF [20], EXPECT [21], Trellis (EXPECT's successor) [22], COGITO [23], TKAI [24], OntoKick [25], etc. (knowledge acquisition from experts and tacit knowledge revealing); OntoEdit [26], Protg [27], OntoLingua [28], etc. (ontologies engineering); HPKB [29], AKT [30] etc. (KBs organization and development); KRAFT [31], InfoSleuth [32], Observer [33], etc. (knowledge and information integration). The above approaches are targeted at pertinent, clear, recent, correct information and knowledge processing and timely delivering to locations of need for global situational awareness and ability to predict development of going on processes at the level of understanding; and there arises a need for KL [34].

The possible application domains of KL belong to the following areas:

- Large-scale dynamic systems (enterprises) with distributed operations in uncertain and rapidly changing environment, where the information collection, assimilation, integration, interpretation, and dissemination are needed [35, 36].
- Focused logistics operations and/or Web-enhanced logistics operations addressing sustainment, transportation and end-to-end rapid supply to the final destination, where the distributed information management and real-time information/KF to support continuous information and knowledge integration of all participants of the operations are needed [37].
- Markets via partnerships with different organizations, where the dynamic identification and analysis of information sources and providing for interoperability between market participants (players) in a semantic manner are needed [38–40].

For all of the above areas it is possible to describe management systems as an organizational combination of people, technologies, procedures and information/knowledge.

The KL is based on individual user requirements, available KSs, and content analysis in the information environment. Hence, systems operating in this area must react dynamically to unforeseen changes and unexpected user needs, keep up-to-date resource value assessment data, support rapid execution of complex operations, and deliver personalized results to the users/knowledge customers. Here proposed approach to KL is realized through the KF — integration of knowledge from different sources (probably, heterogeneous) into a combined resource in order to complement insufficient knowledge and obtain new knowledge. KF is based on a higher level of abstraction and, therefore, utilizing the same methods/techniques is not possible. Development of this scientific direction went a long way from data fusion (Fig. 1), which arises from multisensor data fusion, in which information from a number of sources is integrated to form a unified picture [42].

3 KSNET-APPROACH

3.1 Knowledge Sources

The following KSs types were identified (Figure 2): (i) experts, who directly enter knowledge related to user request using built-in mechanisms, (ii) KBs, (iii) databases, (iv) structured documents — text, HTML, XML, etc. documents (relevance of a document to a request can be estimated using indexed keywords) and (v) other sources, for which mechanisms of knowledge recognition and capturing are available.

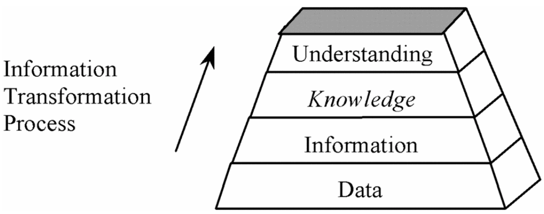


Fig. 1. Conceptual framework of information support (adapted from [41])

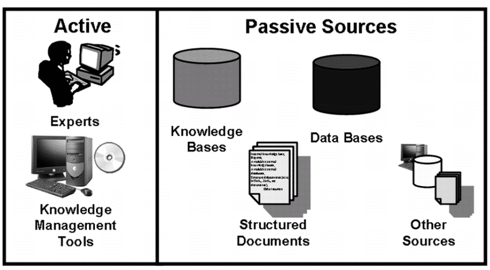


Fig. 2. Knowledge sources

KSs fall into two groups: (i) passive sources (available external data and KBs, structured documents, other sources with some developed mechanisms of interaction) providing knowledge “by demand”; and (ii) active sources (experts, KM tools) providing knowledge ”by demand” and pro-activity functions “Just-Before-Time”-support for request processing.

3.2 Knowledge Fusion

In [43] the most complete sequence of main operations for KM referred to as knowledge chain was proposed. It was used as a basis for the development of KF process structure (Figure 3) consisting of: (i) capturing knowledge from KSs and its translation into a form, suitable for a supplementary use, (ii) acquisition of knowledge from external sources, (iii) selection of knowledge from internal sources (local KBs), (iv) knowledge generation: producing knowledge by discovering or deriving from existing knowledge, (v) internalization: changing system knowledge by saving acquired, selected and generated knowledge, (vi) externalization: embedding knowledge into system's output for release into the environment, (vii) KF management: planning, coordination, collaboration, and control of operations constituting the KF process. In Section 0 the user processing scenario using most of these operations is presented.

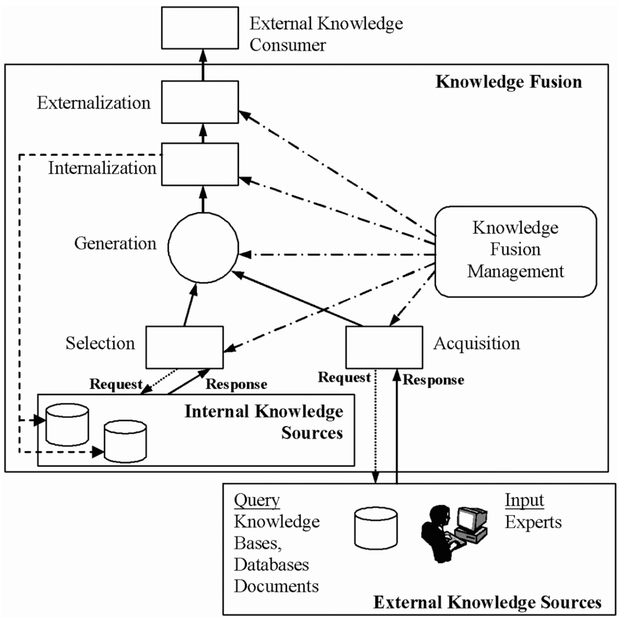


Fig. 3. Operations of the knowledge fusion process

To increase the KF rapidity it is necessary not only to find required sources but also to identify their usefulness for solving a particular problem. For this purpose it is reasonable to: (i) use user profile (structured information about the user), (ii) offer tips and hints to the user to reveal tacit user interests, (iii) utilize techniques of knowledge/ontology reuse, (iv) perform indexation of stored knowledge, and (v) increase intelligibility of knowledge representation for the users, involved into the processes of development, edition, update, etc.

3.3 Knowledge Source Network

Network of loosely coupled sources located in the information environment is referred to as "Knowledge Source Network" (KS network). The term KS network originates from the concept of virtual organization based on the synergistic use of knowledge from multiple sources. Figure 4 explains roughly the basic concepts of the KS networks and their multi-level configuration. The upper level represents a customer-oriented knowledge model based on a fusion of knowledge acquired from KS network units (KSs), which constitute the lower level and contain their own knowledge models.

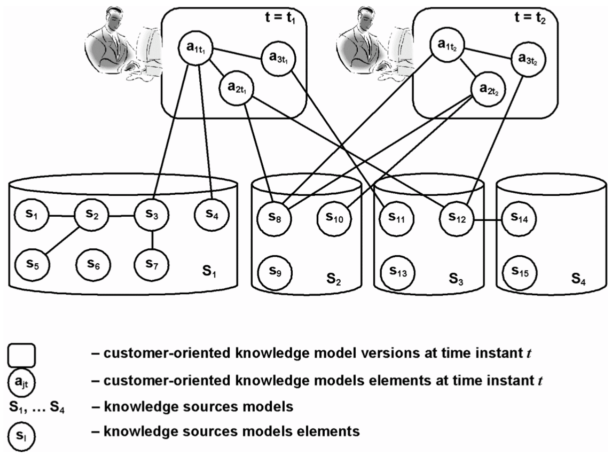


Fig. 4. Distributed multi-level knowledge fusion management as the KS network configuration

3.4 Ontology-Driven Methodology for Knowledge Fusion

The following ontologies types for the KSNet systems were defined: (i) top-level ontology describes notation for application domain description, (ii) application ontology (AO) contains terms and structure of knowledge describing a particular application domain, (iii) preliminary KS ontology (KSO) contains KS' knowledge terms and structure in the top-level ontology notation, (iv) KSO contains correspondence between terms of KS and AO, (v) preliminary request ontology contains terms which can be used by a user for request input and structure in the top-level ontology notation, (vi) request ontology contains correspondence between terms of preliminary request ontology and AO. The ontologies are stored in the common ontology library (OL) that allows reusing them.

The system works in terms of a common vocabulary. AO is based on domain, tasks and methods ontologies also stored in OL. Each user/user group works in terms of associated expandable request ontology and thereby with a part of AO pertinent to the user/ user group. *User profiles* are used during interactions to provide for an efficient personalized service. Every user request consists of two parts: (i) structural constituent (containing the request terms and relations between them), and (ii) parametric constituent (containing additional user-defined constraints). For the request processing, an auxiliary KS network configuration is built defining when and what KSs are to be used for the request processing in the most efficient way. For this purpose a *knowledge map* (see Section 0) including information about locations of KSs is used. Translation between the system's and KS' notations and terms is performed using KSOs. A conceptual scheme of the user-oriented ontology-driven KF methodology is presented in Figure 5.

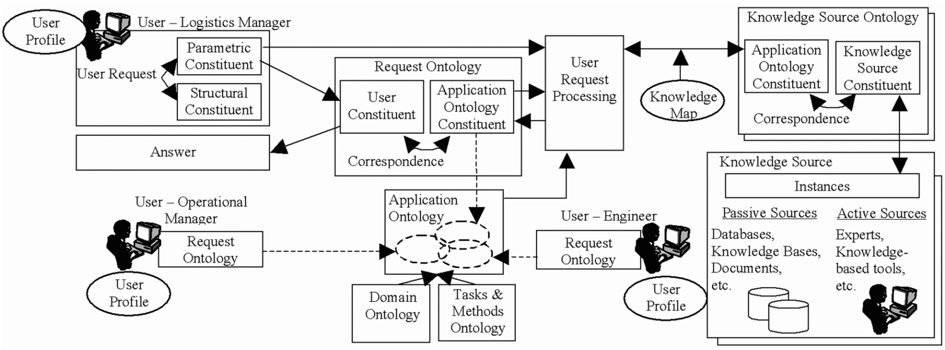


Fig. 5. Conceptual scheme of the user-oriented ontology-driven KF methodology

A formalism of object-oriented constraint networks has been chosen for the ontology representation. An abstract KS network model is based on this formalism. This solution was mainly motivated by such factors as support of declarative representation, efficiency of dynamic constraint satisfaction, and problem modelling capability, maintainability, reusability, and extensibility of the object-oriented technology.

According to the chosen formalism an ontology (A) is defined as:

$$A = (O, Q, D, C), \tag{1}$$

where
O — a set of *object classes* ("*classes*"). Each of the entities in a class is considered as an *instance* of the class. This set consists of two subsets:

$$O = O^I \cup O^{II}, \tag{2}$$

where
O^I — a set of *non-primitive* classes i.e. classes which can have instances:

$$O^I = \{o : \exists \text{ instance } (o)\} \tag{3}$$

O^{II} — a set of *primitive* classes i.e. classes which cannot have instances:

$$O^{II} = \{o : \neg\exists \text{ instance } (o)\} \tag{4}$$

Q — a set of class attributes ("*attributes*")
D — a set of attribute domains ("*domains*")
C — a set of *constraints*. For chosen notation the following six types of constraints have been defined:

$$C = C^I \cup C^{II} \cup C^{III} \cup C^{IV} \cup C^V \cup C^{VI}, \tag{5}$$

where
C^I — accessory of attributes to classes:

$$C^I = \{c^I\}, c^I = (o, q), o \in O, q \in Q \tag{6}$$

C^{II} — accessory of domains to attributes:

$$C^{II} = \{c^{II}\}, c^{II} = (o, q, d), o \in O, q \in Q, d \in D \tag{7}$$

C^{III} — classes compatibility (compatibility structural constraints):

$$C^{III} = \{c^{III}\}, c^{III} = (\{o\}, \text{True} \vee \text{False}), |\{o\}| \geq 2, o \in O \tag{8}$$

C^{IV} — hierarchical relationships (hierarchical structural constraints) "is a" defining class taxonomy (*type* = 0), and "has part"/"part of" defining class hierarchy (*type* = 1):

$$C^{IV} = \{c^{IV}\}, c^{IV} =< o', o'', \text{type}, o' \in O, o'' \in O, o' \neq o'' \tag{9}$$

C^V — associative relationships ("one-level" structural constraints):

$$C^V = \{c^V\}, c^V = (\{o\}), |\{o\}| \geq 2, o \in O \tag{10}$$

C^{VI} — functional constraints referring to the names of classes and attributes.

$$C^{VI} = \{c^{VI}\}, c^{VI} = f(\{o\}, \{o, q\}) \rightarrow \text{True} \vee \text{False}, |\{o\}| \geq 0, |\{q\}| \geq 0, o \in O, q \in Q. \tag{11}$$

The most abstract class of the ontology (the top of the ontology's taxonomy) is "Thing".

$o \in O$, $q \in Q$, $d \in D$, $c \in C$ are considered as ontology elements.

4 KNOWLEDGE FUSION SYSTEM KSNET

4.1 Knowledge Fusion System Organizational Principles

In a result of the analysis of the modern systems for KM and information/knowledge fusion [31–33], [44, 45], the major organizational principles of the KSNet system based on KSNet-approach have been formulated as follows: (i) scenarios and procedures of the system are developed independently on application domain; (ii) the system must deal with a specific application domain; (iii) the system must provide interface for request input and result representation; (iv) the system must perform translation of the entered request into application domain terms, decomposition of the request into its components — subrequests, recognize the subrequests and send them to processing (identifying suitable KSs and creating a special configuration of the KS network, querying identified sources, filtering them according to user-defined constraints, fusion of knowledge from different sources, validation, check for meeting requirements, presenting results to the user), (v) subrequests are processed simultaneously, (vi) the request can be passed to experts specializing in the application domain, (vii) results must be recorded, internal information components of the KSNet system have to be changed for supplementary reuse in similar requests.

4.2 Main Components

In accordance with the above organizational principles the following components of the KSNet system were identified:

1. Software components: (a) methods, (b) agents, (c) interface for user request input, for new knowledge entry by an expert, for operations with application domain (import and creation of AO, searching and ranking of KSs, preparation of special interface forms (request templates) for knowledge customers and administrators), for OL support (ontologies import and export, maintenance, diagnostics), etc.
2. Repository: (a) ontologies (top-level ontology, AO, preliminary KSO, KSO, request ontology, and preliminary request ontology) and (b) information components (internal KB, knowledge map, user profiles).
3. Service tables.

The components are related to each other. Information about all elements the system deals with (KSs, experts, users, tools, etc.) and main terms describing the application domain are stored in the service tables, and all active elements (users, active KSs, etc.) access them via specially designed software interfaces. Service tables are created and maintained via a database management system (DBMS).

Service tables are meant to store information describing application domain, user profiles, parameters of KSs, namely: (i) contents of ontologies, user profiles, internal KB, knowledge map, (ii) links to methods and KSs, and (iii) information about the KSNet system users, wrappers, auxiliary reference information.

Figure 6 represents a scheme of the KSNet system components. Solid arrows show that components' content is stored in the service tables; dashed arrows show that the service tables contain references to the components. Although KSs are not the components of the KSNet system, they are also represented in the figure since information about them is stored in the service tables. Information about the users is accessed through “Interface” and “User Profile” components.

Figure 7 presents the KSNet system repository structure. The following three components in the repository structure were defined. A semantic component is used for knowledge representation in a common notation and terms.

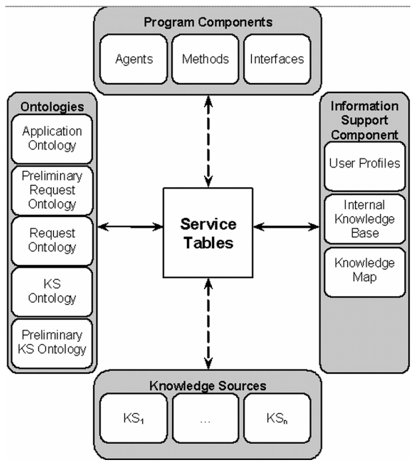


Fig. 6. Environment components

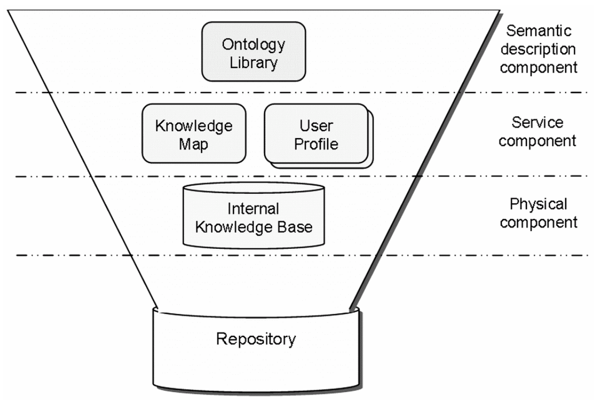


Fig. 7. The KSNet system repository structure

A service component is used for knowledge indexing and search and contains the following components:

- Knowledge map includes information about locations of KS network units utilized during problem solving and information about alternative sources (KS network units) containing similar information and KSs characteristics. Monitoring tools perform permanent checking of KSs availability and perform appropriate changes in the knowledge map. Knowledge map is meant to facilitate and speed up the process of the KSs choice.

- User profile is an organized storage of information about the user, his (her) requests history, etc. This component is used for a number of purposes (faster search due to analyzing and utilizing request history and user preferences, Just-before-Time request processing, etc.).

A physical component contains internal KB, used for storage, verification and reuse of knowledge (i) entered by experts, (ii) learned from users (knowledge consumers), (iii) obtained as a result of the KF process, (iv) acquired from KSs which are not free, not easily accessible, etc.

4.3 Multi-Agent Architecture

Like some other KM systems, the KSNet system uses intelligent software agents to provide access to distributed heterogeneous KSs [46–48]. Table 1 describes some special features of the agents, used in the KSNet system. Multi-agent system architecture, based on Foundation for Intelligent Physical Agents (FIPA) Reference Model [49] as an infrastructure for definition of agent properties and functions, was chosen as a technological basis for the KSNet system since it provides standards for heterogeneous interacting agents and agent-based systems, and specifies ontologies and negotiation protocols to support interoperability in specific application areas. FIPA-based technological kernel agents used in the system are: wrapper (interaction with KSs), facilitator ("yellow pages" directory service for the agents), mediator (task execution control), and user agent (interaction with users). The following problem-oriented agents specific for KF, and scenarios for their collaboration were developed: translation agent (terms translation between different vocabularies), KF agent (KF operation performance), configuration agent (efficient use of KSNet), ontology management agent (ontology operations performance), expert assistant agent (interaction with experts), and monitoring agent (KSs verifications).

A major set of agents is represented in Figure 8 according to the principles and functions of the KSNet system described above.

Each agent of the KSNet system contains the following modules [50]: (i) identifying, (ii) functional, and (iii) repository. Identifying module contains such parameters as unique identifier, creation date and time, etc. This module structure depends on agent type (some agents do not need this module). Functional module contains a set of procedures to be executed by the agent. Repository contains special information, such as agent's knowledge, history of the agent's contacts, temporary results, new knowledge, etc. Identifying and functional agent modules are shown in Figure 9. The agents' connectivity matrix is presented in Figure 10.

Agent	Life time	Quantity	General tasks
Wrapper	KSs life time	Number of KSs types	Translates knowledge from source terms into the AO terms and sends requests from system to sources.
Mediator	Task execution time	Number of tasks being processed	Tracks out task processing step-by-step from input to result. Provides negotiations with the expert assistant agents during alternative KS ranking. Stores temporary results.
Facilitator	System life time	1	Provides a "yellow pages" directory service for the agents.
User agent	As long as user is registered in the system	Number of registered users	Provides user personalization service: (provides a set of functions for the user profile processing, facilitates request input, provides a set of tips and hints for the user, and passes messages and information from the system to the user.
Translation agent	System life time	Number of request input interfaces	Provides for translation of terms between the users and the system, between application domain and KSs. Uses the request ontology, and AO.
Expert assistant agent	As long as the expert is registered in the system	Number of registered experts	Facilitates the process of expert knowledge entry into the system. Supports the process of alternative KS ranking. Updates the user profiles.
Configuration agent	System life time	1	Configures KS network using the knowledge map and the user profiles. Performs scheduling functions. Negotiates with the KF agents and the wrappers.
Knowledge fusion agent	System life time	Varying	Obtains knowledge from the mediator and processes it. Generates new knowledge. Validates it. Interacts with the monitoring agent.
Monitoring agent	System life time	1	Provides a set of functions for diagnostics of the system repository base and external KSs.
Ontology management agent	System life time	1	Provides a set of functions for ontology engineering and operation — creation of ontologies for new KSs, modification of AO etc. Checks correspondence between KS and request ontologies and AO.

Table 1. Features of agents of the KSNet system

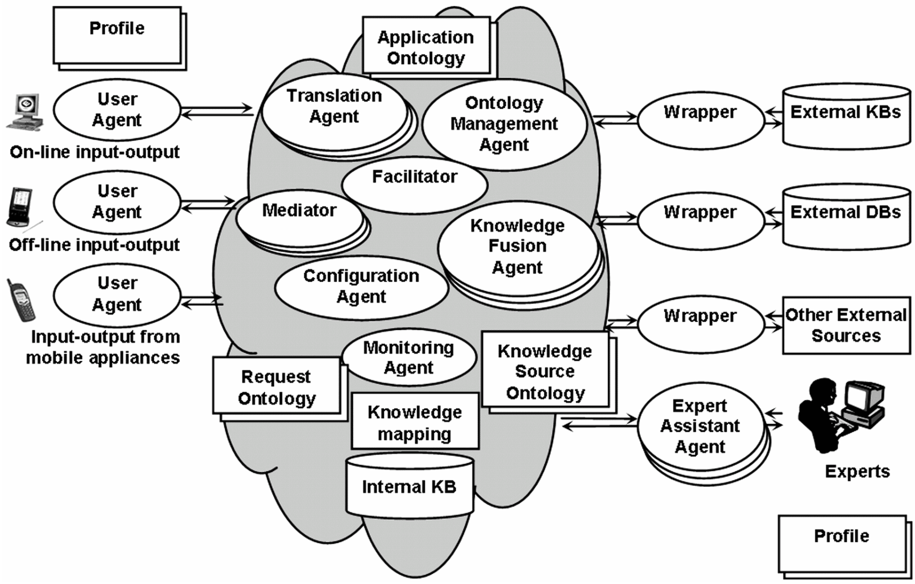


Fig. 8. Basic components of the multi-agent KSNet system

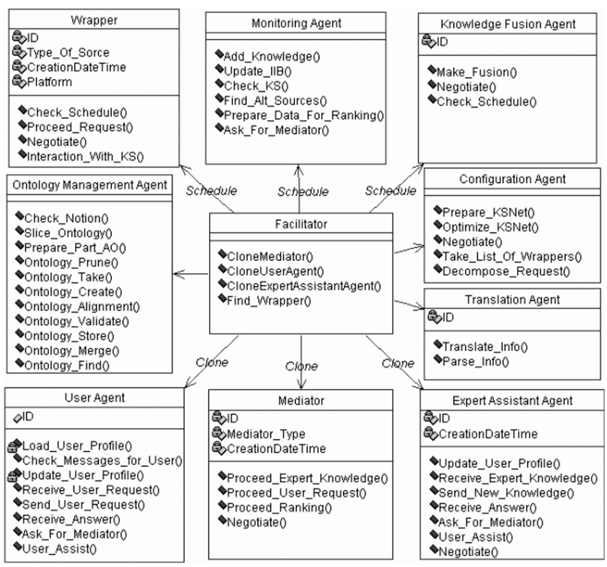


Fig. 9. Class diagram of KSNet agents with main properties and functions

Caller \ Callee	Wrapper	Mediator	Facilitator	User Agent	Translation Agent	EA Agent	Configuration Agent	KF Agent	Monitoring Agent	OM Agent
Wrapper		P	P		P					P
Mediator	P			P	P	P	P	P	P	
Facilitator		P								
User Agent		M/P	P							
Translation Agent										P
Expert Assistant (EA) Agent		M/P	P							
Configuration Agent	M/P	P	P					M/P		
KF Agent		P	P						P	
Monitoring Agent		M/P								
Ontology Management (OM) Agent	P				P				P	

Fig. 10. Agents' connectivity matrix (P — peer-to-peer interaction, M — mediating interaction)

4.4 Users of the KSNet System

A list of main functions of the KSNet system users and agents interacting with users to perform these functions are given in Table 2. The users are divided into two groups: (i) users serving the system and (ii) users served by the system (knowledge customers).

At the early stage of the system's operation, knowledge consumers and experts pass the registration procedure. This process assumes entry of initial user information by filling-in a form or a questionnaire, generating the user profile, granting user rights. Later the user passes the authentication (user rights verification) process only.

Administrator is a user constantly working with the KSNET system. All other users are engaged at some stages of the system preparation and operation. Ontology engineers work together with experts and software engineers. Since knowledge acquisition from an expert is a complex task, knowledge engineers are required to facilitate this process [51].

Agents support (i) work of *ontology engineers*: help them to import and build ontologies, (ii) the process of direct knowledge entry by *knowledge engineers* into the *internal knowledge base*, (iii) the process of alternative KSs ranking by *experts*, and (iv) the process of *knowledge customer* requests processing. They provide administrator the results of KS monitoring. The *software engineers* interact with agents indirectly — they develop and attach methods which are used by the agents and take part in the development of the *wrappers* for new types of KSs.

User Types	Main Functions	Agents
Administrator	• service table processing (archiving, indexing, etc.); • backup creation; • system diagnostics; • diagnostics reports processing; • user rights management.	• Monitoring agent
Experts	• application domain studying, • KSO creation, • AO creation, • search for KSs, • alternative KSs ranking (regarding the domain). • new knowledge entry, • alternative KSs ranking (regarding the stored requests).	• Monitoring agent • Mediator • Facilitator • Expert assistant agent • User agent
Knowledge consumers	• input of requests ("user requests").	• User agent • Mediator • Monitoring agent • Facilitator • Translation agent • Ontology management agent • Wrapper • Configuration agent • KF agent
Software engineers	• adaptation, creation, and connection of methods for task solving, parsing and translation of attribute values obtained from KSs; • connection of new KSs and KS types.	All agent types
Ontology engineers	• AO creation; • preliminary KSO and KSO creation; • request ontology creation; • modification of ontologies stored in ontology library.	• User agent • Ontology management agent • Monitoring agent • Facilitator • Mediator
Knowledge engineers	• internal knowledge base validation; • search for missing knowledge.	• User agent • Expert assistant agent • Mediator • Monitoring agent • Facilitator • Translation agent • Ontology management agent

Table 2. The KSNet system users

4.5 Major Scenarios

The KSNET system life cycle consists of two major phases:

- *Preparation phase*. It includes the following tasks: (i) study of application domain, creation of AO based on existing OL and describing the application domain; (ii) search for KSs related to the application domain and creation of KSOs and (iii) configuration of the KS network — knowledge distribution within KSs;
- *Operation phase*. It includes the following tasks: (i) offer of end-user interface for entry of knowledge search request by knowledge consumer; (ii) selection of KSs related to user request and configuration of KS network; (iii) selection, acquisition, fusion and verification of acquired and generated knowledge; (iv) presentation of results to the user; (v) storage of the results. The configuration of the KS network consists of: (i) selection of KSs which are to be included in KS network; (ii) negotiation between the KS network units; and (iii) scheduling and coordination of the KS network.

Below, one of the major KSNet system scenarios of user request processing during the operation phase (Figure 11) is presented. When a user request is received by the KSNet system, its terms are translated into the system terms using the request ontology. Based on the translated request a part of AO is formed which describes an object-oriented constraint network for user request processing. This object-oriented constraint network is a basis for building requests to KSs. A request to KS is translated into the terms (using KSO) and notations (using preliminary KSO) of KS. An answer from KS is translated back into the notation and terms of the KSNet system and passed to the constraint network of the user request processing. The results of the processing are analyzed by the system and can be added to the internal KB for their possible reuse and/or in AO. The user request processing is completed by translation of the request processing results into the user terms and presenting them to the user. All the translation operations are performed using appropriate ontologies.

This step by step scenario supported by multi-agent architecture is presented below (Figure 12).

1. The *user* (*knowledge customer*) interacting with the *user agent* enters the request into the KSNet system.
2. The *user agent* asks the *facilitator* for a *mediator*, which will trace this task. The *facilitator* clones the *mediator* and assigns the task to it.
3. The *user agent* makes appropriate changes in the *user profile*.
4. The *mediator* receives the request from the *user agent*.
5. The *translation agent* interacting with the *ontology management agent* recognizes terms of the user request and returns translated request to the *mediator* (knowledge internalization operation).

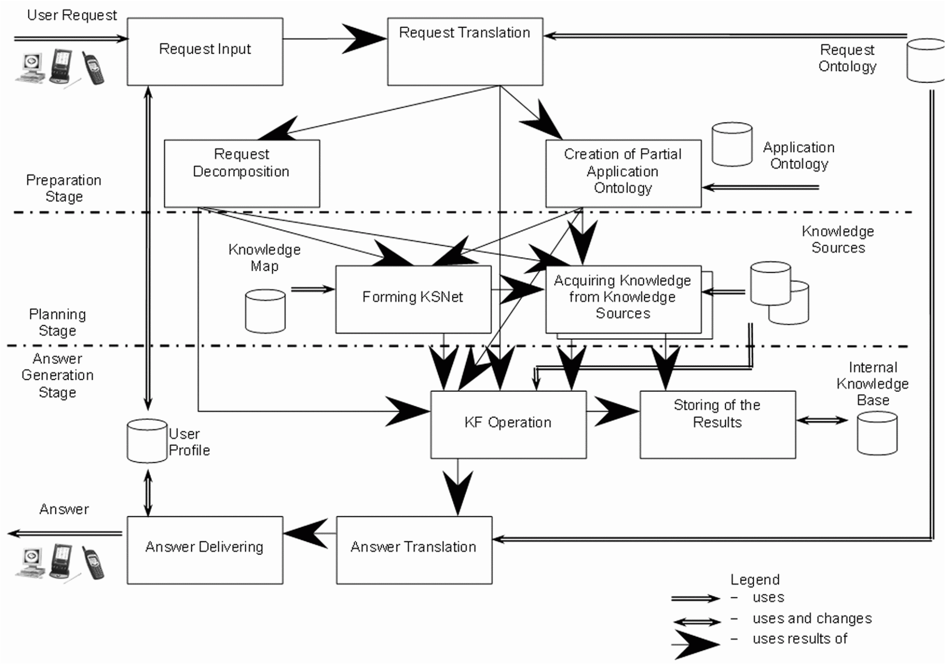


Fig. 11. Collaboration diagram of user request processing scenario

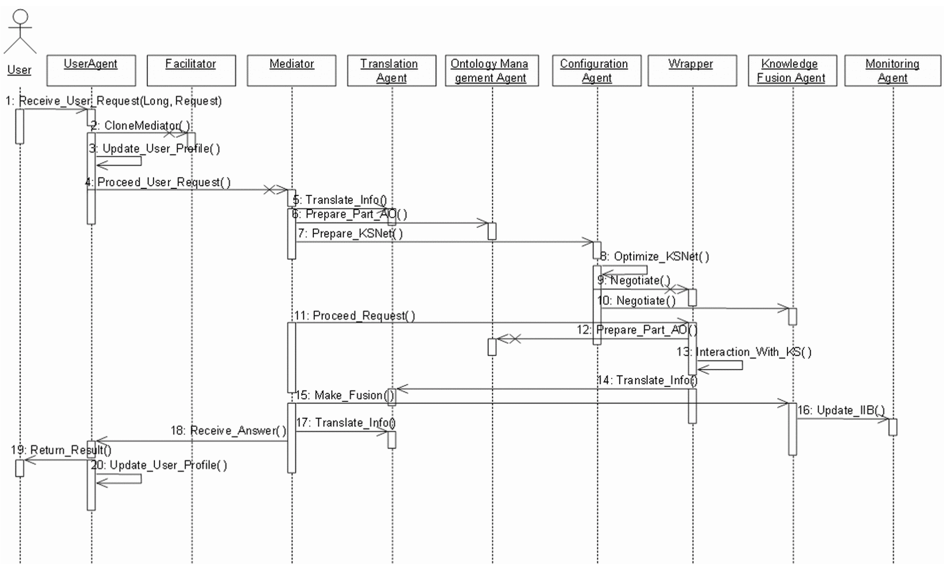


Fig. 12. Sequence diagram of user request processing

6. The *ontology management agent* forms the *request ontology* for the user, and slices a part of the AO corresponding to the user request.
7. The *configuration agent* searches for similar requests in the *knowledge map* and the *user profiles*, performs decomposition of the request into subrequests, identifies appropriate KSs (Section 5.4).
8. The *configuration agent* starts the process of KS network for user request configuration.
9. The *configuration agent* plans KF by negotiating price, schedules, capabilities, etc. with the *wrappers*.
10. The *configuration agent* plans KF by negotiating time, capabilities, etc. with the *KF agent*. The *configuration agent* passes results of KS network configuration to the *mediator*.
11. The *mediator* passes the subrequests to the *wrappers* according to the KS network.
12. The *wrappers* pass subrequests to the *ontology management agent* for translation into the KSs terms (knowledge externalization operation), transform them to the KSs notation, perform control for error, malfunction or failure occurrence.
13. The *wrappers* pass the subrequests to the KSs, receive response from the KSs (knowledge capturing operation) and transform the response into the system's notation.
14. The *wrappers* pass the answers to the *translation agent* for translations into system's terms (knowledge internalization operation) and return the result to the *mediator*.
15. The *mediator* passes the results received from the *wrappers* to the *KF agents*. The *KF agents* perform fusion of received knowledge (knowledge generation operation), validate new knowledge and check it for relevance to the request.
16. The *KF agents* pass the new knowledge to the *monitoring agent* for internal processing. The *monitoring agent* performs appropriate changes in the *system repository*.
17. The *mediator* passes the result of KF to the *translation agent* for processing. The *translation agent* translates the system's result into a form understandable by the user (knowledge externalization operation).
18. The *mediator* passes the results to the *user agent*.
19. The *user agent* returns the results to the user.
20. The *user agent* updates the *user profile*.

4.6 Knowledge Fusion Patterns

In the KSNet systems the process of KF takes place during performance of different tasks. Carried out analysis of major system scenarios has allowed to select a list of generic KF patterns for these operations (Figure 13).

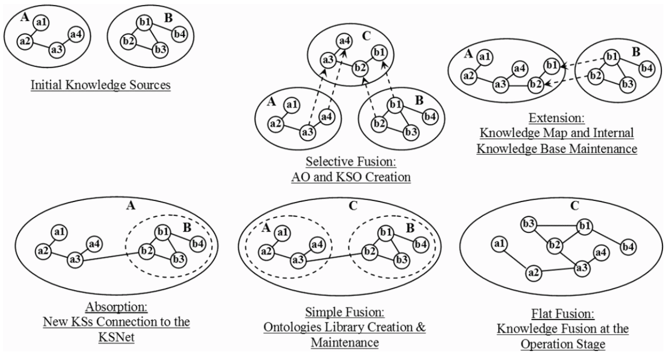


Fig. 13. Knowledge fusion patterns

Definition of the developed KF patterns can be illustrated via the following example. Two initial KSs (A and B) with some structures of primary knowledge units are given. There is a tacit relation between two primary knowledge units, namely a3 from A and b2 from B. It is necessary to fuse two sources preserving the internal knowledge structure and revealing the above tacit relation.

- Selective Fusion (AO and KSO creation). New KS is created, which contains required parts of the initial KSs. Initial KSs preserve their internal structures and autonomy.
- Simple Fusion (OL creation and maintenance). New KS is created, which contains initial KSs. Initial KSs preserve their internal structures and lose (partially or completely) their autonomy.
- Extension (*knowledge map* and *internal KB* maintenance). One of initial KS is extended so that it includes the required part of other initial KS, which preserves its internal structure and autonomy.
- Absorption (a new KS connection to the system). One of initial KSs is extended so that it includes other initial KS, which preserves its internal structure and loses (partially or completely) its autonomy.
- Flat Fusion (KF at the operation phase). New KS is created, which contains initial KSs. The initial KSs dissolve within new KS and do not preserve their internal structures and autonomy.

Based on the definition of the KF patterns different patterns have been chosen for different tasks of the KSNet system (Table 3). The use of the KF patterns allowed to accelerate the KF process due to typification of fusion schemes.

5 PROTOTYPES AND EXAMPLES

The main goal of the case study described below is to test implementation of the KSNet-approach for complex dynamic systems — "product — process — business

Task	Description	KF Patterns
AO and KSO creation during preparation and operation phases	New ontologies are built using elements of existing ontologies or KSs	Selective Fusion
OL creation and maintenance during preparation and operation phases	OL contains different ontologies	Simple Fusion
Knowledge map and internal KB maintenance during operation phase	Knowledge map and internal KB are extended when new KSs are connected to the system	Extension
A new KS connection to the system	The new KS becomes a part of the system from the user point of view	Absorption
KF at the operation phase	Knowledge from different sources is used for generation of new knowledge	Flat Fusion

Table 3. Usage of Knowledge Fusion Patterns

organization (business)" systems — of different configuration types: (i) marketing/order configuration, (ii) product configuration, (iii) upgrade/add-on configuration, (iv) distributed process configuration, (v) business network unit configuration, and (vi) whole business network configuration.

5.1 Distributed Architecture of Prototype

The key points for the project to be tested and prototyped were conditioned by covering all the KF patterns (mainly concentrated on the preparation phase of the KSNet system lifecycle), KS network configuration (as a main element providing for the rapidity of the KF process) and constraint network processing (as a basis technology providing for the user request processing). In accordance with up-to-date technologies and standards the information kernel for KL is built as shown in Figure 14.

As described above, the knowledge is represented by an aggregate of interrelated classes, their attributes, attributes' domains and relations between them. An object scheme for working with the knowledge and database structure for its internal storage are designed based on this notation. An access to the database is performed via ODBC as a standard data access mechanism under MS Windows operation system. Remote access to the stored knowledge is performed via common HTTP Internet protocol. Knowledge representation is done via either interactive HTML+VRML Java enabled pages for users or DAML+OIL-based format for knowledge-based tools.

In order to increase rapidity of the KF process in the KSNet system the following supporting tasks were defined (Figure 15): (i) the knowledge map creation utilizing alternative KSs ranking, (ii) KS network configuration based on the task

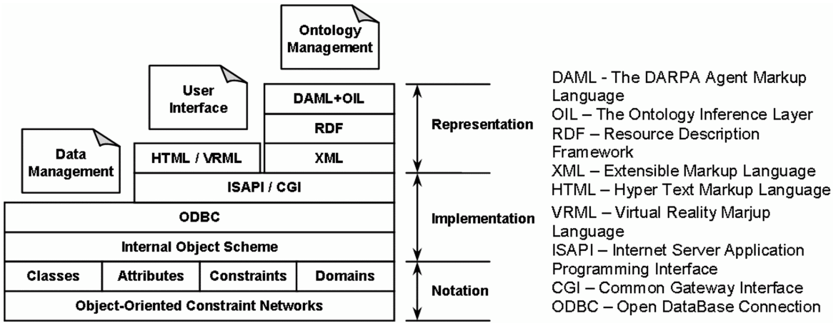


Fig. 14. Standards of knowledge logistics information kernel

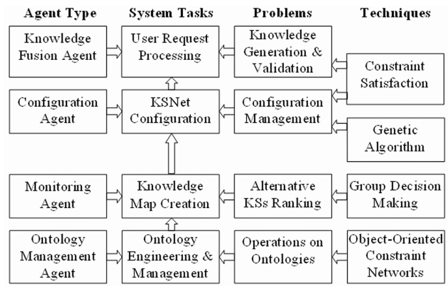


Fig. 15. Main system tasks and techniques

of efficient KSs choice, and (iii) user request processing based on constraint network processing. These tasks require development and application of appropriate mathematical mechanisms (models and methods). The "MultiExpert" system, based on group decision support technique, is used for the knowledge map creation. The application based on genetic algorithm is used for KS network configuration. The

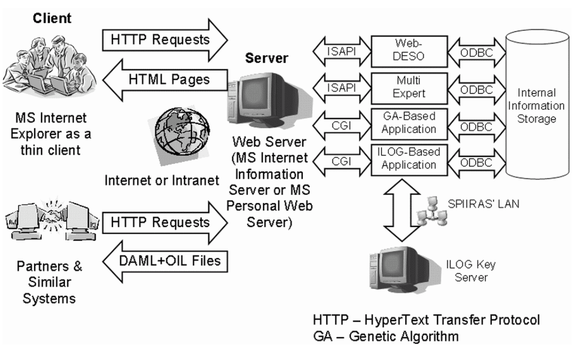


Fig. 16. The architecture of the research prototype of the KSNet system

application based on ILOG constraint satisfaction technology is used for constraint network processing. The architecture of the research prototype of the KSNet system is presented in Figure 16.

5.2 "Web-DESO" — the Prototype for Ontology Library Management

OL management is performed by the ontology management agent. The main criteria for implementation of OL were: (i) support of the chosen ontology notation and structure, (ii) compatibility with other formats such as DAML+OIL, and (iii) web-based interface enabling remote collaborative work with ontologies.

Some tools for ontology creation/management (Ontolingua, OilEd, Protégé, etc.) have been tested but none of them met the above requirements. Thereby the client-server architecture of the Web-DESO (Web-DEsign of Structured Objects) has been developed to match the above criteria.

Web-DESO works with one OL containing several ontologies (Figure 17). An ontology can be one of the following types: domain ontology, tasks & methods ontology, AO, KSO, request ontology. Ontology management environment supports ontology taxonomy view and allows creating and editing classes, attributes and constraints of accessory of attributes to classes, domains for class attributes, class compatibi-

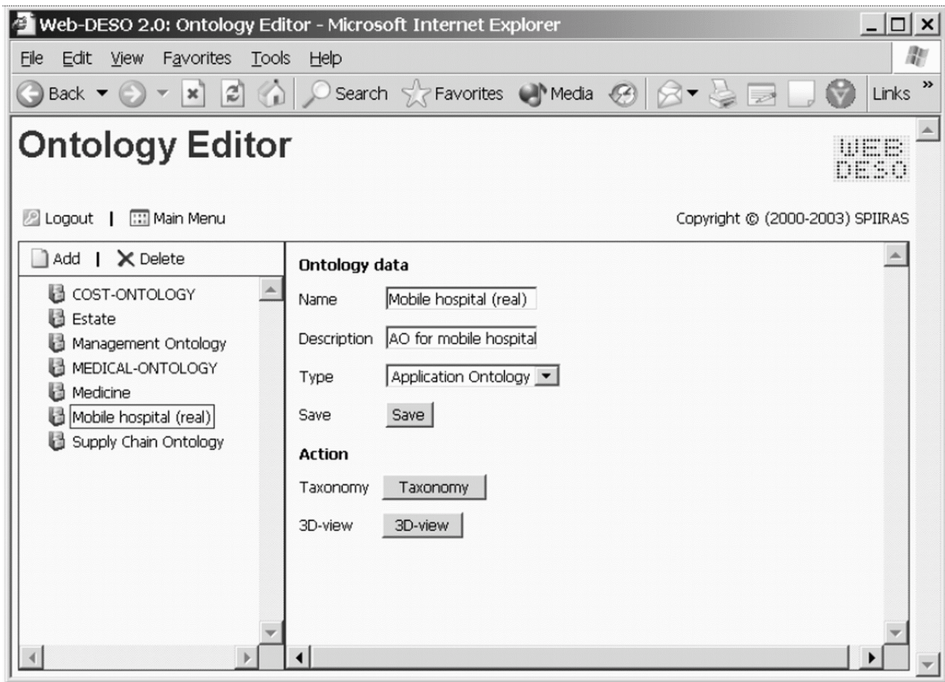


Fig. 17. Ontology view of Web-DESO

lity, taxonomy, and associative. In contains some additional modules: (i) "Term searcher" to find ontologies and their elements by keywords, (ii) "Ontology merger" to build new ontology using parts of existing ontologies, (iii) "Ontology importer" to import ontologies from DAML+OIL format and some service modules.

The comparison of Web-DESO system with some other ontology management tools is presented in Table 4. In order to ensure that the designed prototype is compatible with other standards a mechanism of import/export of ontologies described in DAML+OIL has been developed and implemented.

5.3 "MultiExpert" — the Prototype for Alternative Knowledge Source Ranking

In accordance with the KSNet system architecture suggested above the monitoring agent defines a set of alternative KSs to be evaluated. For this purpose it reads knowledge map and user profiles revealing necessary KSs' parameters and preparing required auxiliary data. Initial data for a group of experts is prepared in the form of a task, which contains a list of evaluative and classifying parameters (criteria for KSs evaluation) and descriptions of KSs. The list of parameters is fixed (knowledge volume, on-line access, correctness, explanation quality, intelligibility, access bandwidth, request processing time, popularity, reusability, and possibility to be changed), but can be modified.

To order and to rank alternative KSs, "MultiExpert" tool is used [52]. The monitoring agent forms a group of experts who solve the task, and develops personal calendars of the tasks for the experts. For this purpose it reads user profiles studying accumulated knowledge about available experts, their preferences, schedules, commitments, etc. When the group is created, the monitoring agent prepares necessary auxiliary tables for experts' work support and asks the facilitator to create a mediator for the task processing. The facilitator clones an instance of the mediator which is supposed to interact with the monitoring agent and expert assistant agents.

The expert assistant agents gather results of experts' work and pass them to the mediator. The mediator makes a group decision, negotiates with the expert assistant agents for group decision conforming. To facilitate the organization of collaborative decision making a method utilizing decision theory techniques [53] is used. The method produces conformed decision based on iterating procedure of local decisions aggregation to come into an agreement with that of a group being processed. To obtain conformed agents' decisions two levels of abstraction are distinguished: local level (associated with the expert assistant agent) and group level (associated with the mediator), aggregating local opinions. A feedback is needed at both steps of the algorithm: group decision aggregation or initial data representation. To obtain conformed decision (i) agents iteratively change utility functions or (ii) group estimation is involved as an additional attribute. A wholesome function is used to choose a decision.

When the task passes all the stages it is considered as finished, i.e. all alternative KSs have the assigned weights. These values are passed to the KSNet system and

Principles	Tools				
	Ontolingua	Protégé	OilEd	OntoEdit	Web-DESO
Creators, projects	Stanford University Knowledge Systems Laboratory	Stanford University Medical Informatics Laboratory, Semantic Web Project	The University of Manchester, On-To-Knowledge-Project	Ontoprise GmbH, Semantic Web Project	SPIIRAS, KSNet-approach for knowledge fusion
Knowledge representation formalism	Monotonic first-order logic	OKBC-oriented frame-based model	Description logic	RDF-oriented frame-based model	Object-oriented constraint networks
Methods for modelling concepts and relations	A set of axiomatized taxonomies with relations among them	Taxonomy	Taxonomy, hierarchy	Taxonomy, hierarchy	Taxonomy, hierarchy, associative relationships
Knowledge role-limiting methods	Axiomatic logic theories	Axioms, constraints	Rule base	Axioms as first order entities	Classes compatibility, functional relationships
Currently supported formats	Loom, KIF, CycL, Express	RDF	RDF/ RDF-S, DAML+ OIL	DAML+ OIL, FLOGIC, SQL2	DAML+ OIL
Possibilities of ontology reuse	Inclusion, restriction, polymorphic refinement	Inclusion, merging	Inclusion	N/A	Merging
Implementation	Client-server architecture: Server – N/A Client – HTML	Desktop Java-application	Desktop Java-application	Desktop Java-application	Client-server architecture: Server – IIS, ISAPI, PHP, MS Access Client – HTML, Java scripts

Table 4. Comparison of ontology management tools

stored in the knowledge map. These weights are to be used during configuration of the KS network for estimation of KSs quality, reliability, etc.

5.4 Utilizing Genetic Algorithms for Knowledge Source Network Configuration

Knowledge map plays an important role in the rapid KF process. It provides a service of knowledge indexing: stores (i) correspondence between AOs elements and KSs, and (ii) KSs characteristics. Two or more alternative KSs can correspond to one pair class and attribute from AO. Knowledge can be acquired from any of this KSs meeting user defined constraints included in possible KS network configuration.

The goal of this task is a selection by the configuration agent of KSs which can be used for user request processing in the most efficient way according to the predefined criteria such as costs and/or time. The task of efficient KSs choice can be defined as a configuration of feasible (in accordance with a given set of structural constraints) and efficient (in accordance with a given criteria) KS network and definition of a set of rules prescribing when to use a certain KS.

For this task a number of solution techniques were tried and a Genetic Algorithm (GA) was chosen [54] as a probabilistic approach to pseudooptimal solutions search. It suited best for the task of the enumeration nature. Among other techniques such as k-nearest neighbor (this technique was computationally intensive for large data sets) and decision tree (this technique did not provide satisfactory reduction of the decision space) are worth mentioning.

Initially a random set of solutions is generated, then solutions are estimated; the set is sorted according to the chosen criteria, and mutation mechanism is applied to the best solution to generate new solutions. Newly generated solutions are sorted, and new iteration is performed (Figure 18). The process is stopped after a predefined number of iterations.

To investigate the efficiency of GA a set of experiments with a basic GA for tasks of different dimensions have been performed, with KSs' parameters and knowledge maps being randomly generated. The results indicate that the number of required calculations for obtaining a quasi-efficient decision even using basic non-optimized GA is smaller than that in the exhaustive search method. Figure 19 represents the ratio of calculations number for the exhaustive search method to that for the GA, and this improvement grows nonlinearly along with the task dimension growth.

5.5 Utilizing ILOG for Knowledge Fusion in Configuration Task

For implementation of constraint networks utilizing the features of ILOG Configurator [55] a package is proposed representing the task in the object-oriented form (Figure 20).

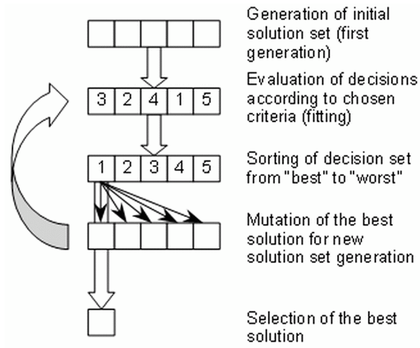


Fig. 18. Genetic algorithm implementation scheme

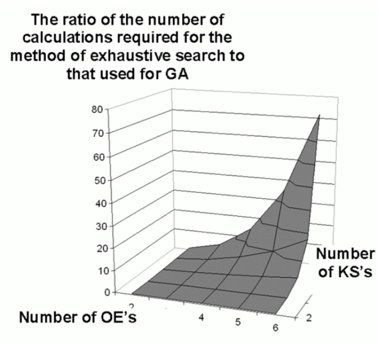


Fig. 19. Efficiency improvement due to GA application

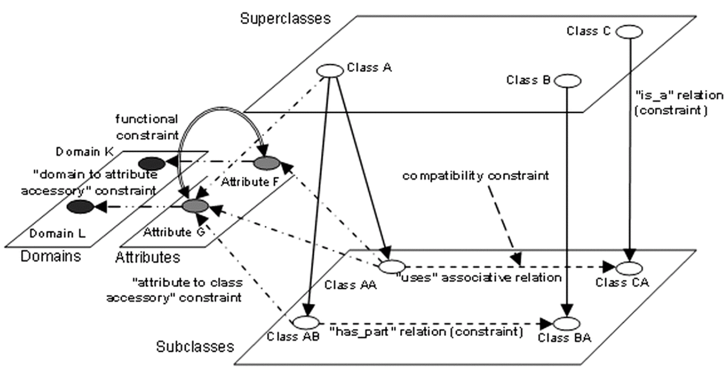


Fig. 20. ILOG Constraint Satisfaction Model adapted to the knowledge representation formalism

In order to verify application of ILOG for KF the following two prototypes of configuration tasks were developed: (i) resource allocation example — a production system for car assembly and (ii) product configuration example — a car configuration. The main idea of the system configuration task (a car in the example) is to obtain a feasible configuration of a system meeting specified requirements, with system structure being known. The task of resource allocation assumes that there is work to be done and some facilities which can perform this work are available. The work consists of several operations (parallel and/or sequential) and each facility is capable to perform some of the operations.

During the preparation stage an ontology engineer creates the AO "Supply Chain Management" for configuration task solving. A supply chain consists of production units capable to perform a number of operations (Figure 21). Every component (node) is described as a set of attributes/properties and a set of possible solutions/templates. Both products and units are described in a domain ontology.

In this example the system's OL contains two domain ontologies ("Management", and "Supply Chain"), and tasks and methods ontology (Figure 22). It is necessary to create "Supply Chain Management" ontology. In the figures presented below the hierarchical relationships ("part of") are shown as solid lines and associative relationships ("uses") are shown as dashed lines. Arrows denote references to tasks/methods (shown only for "Planning" task in Figure 22 and omitted in other figures). In the given example there are the following keywords for "Supply Chain Management" ontology (shaded boxes): *supply network*, *product*, *unit*, *process*, *resource*, *cost centre*. The figures below contain only classes and relationships between them, while the actual operations are performed on the entire set of ontology elements including classes, attributes, constraints/relationships, and domains.

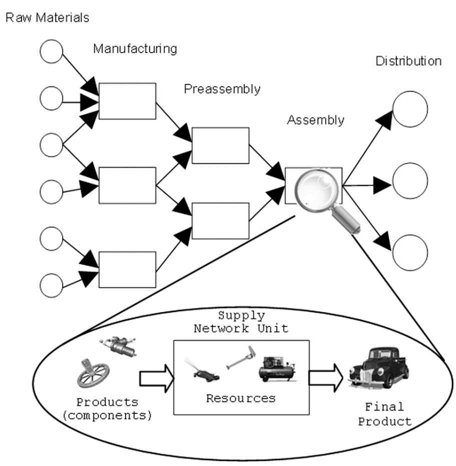


Fig. 21. General scheme of a supply network

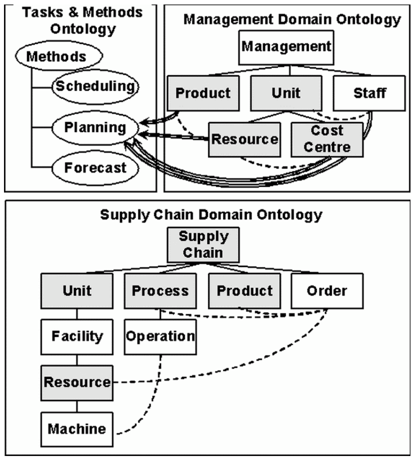


Fig. 22. Ontology library containing task & methods and domain ontologies

During the *Slicing* operation (selecting a portion of an input ontology for use in a new application or new ontology [56]) sets of ontology elements are selected and extracted. The slice of the "Management" ontology is presented in Figure 23; the slice of the "Supply Chain" ontology contains all its elements.

During the *Merging* operation (merging two independently developed ontologies involving resolution of conflicts between term names and structural representations [56]) the slices above are combined into a single set. At the *Pruning* operation (deleting concepts or a sub-hierarchy of concepts that are not needed for a given domain [57]) class "Management" is deleted because of its redundancy: the hierarchy cannot have two roots (Figure 24).

During the *Modifying* operation experts make the following changes (the result is presented in Figure 25): (i) deleting hierarchical relationship connecting "Cost Centre" class with "Unit" class, and adding hierarchical relationship connecting "Cost Centre" class with "Facility" class as required for more precise costs estimation in the current domain, and (ii) deleting classes "Operation" and "Machine" since these classes are beyond the current problem scope (relationships connecting these classes with other classes are deleted automatically).

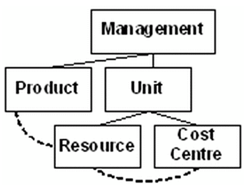


Fig. 23. Slice of the "Management" ontology

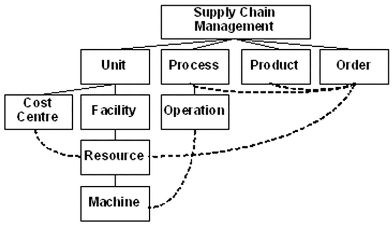


Fig. 24. Resulting set of the *Merging* and *Pruning* operations

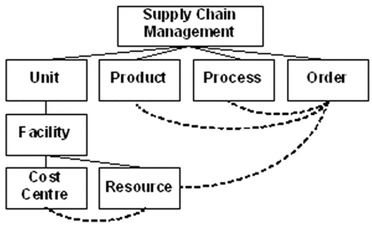


Fig. 25. Supply Chain Management ontology

Since the *Validation* operation (performed automatically and based on predefined conditions and rules) does not show any inconsistencies in the resulting set, it can be referred to as the "Supply Chain Management" AO. This ontology will be used in the example described below.

In the KSNet system users can input their requests in two main ways: (i) in a free form, and (ii) using specially developed templates (dynamic software forms with fields for entering request terms, constraints and criteria). Requests entered in a free form are passed to the system as a plain text, e.g.: "Configure a supply chain (SC) in the part of car components production allocation to SC participants (SC participants are to be found). In accordance with the order the production costs must be minimal, car engine volume must be 2.0 l, and the total cost must be equal or less than $ 25000". Then this request is recognized by the translation agent. The ontology management agent finds correspondence between ontology elements and user request terms. The configuration agent extracts information from knowledge map; using GA it finds KSs containing information for user request processing. It negotiates with wrappers (price, schedules, capabilities, etc.), defines appropriate KSs and prepares KS network configuration for the user request. The wrappers define parts of the request related to their KSs and pass them to the KSs, receive response from the KSs, transform the response into the system's notation. The KF agent performs fusion of received knowledge: it prepares input data for ILOG Configurator and calls its functions and receives answer. The user agent returns the results to the user and updates the user profile.

Utilizing request templates makes recognizing of the requests easier for the system (the processes of request recognition by the translation agent and ontology finding by the ontology management agent are omitted). In order to build such templates a machine learning technology is used to analyze a history of user requests, discovering common patterns and similar requests. Figure 26 presents an example sequence of templates for user request input and answer representation considering a configuration of a car and a supply chain for its production in accordance with user's preferences (based on the free form request given above). The production process consists of three parallel tasks: (i) body production, (ii) engine production, and (iii) transmission production. Facilities are the plants, with known capacities and such characteristics as production cost and time. The goal is cost minimization within time limit or time minimization within cost limit.

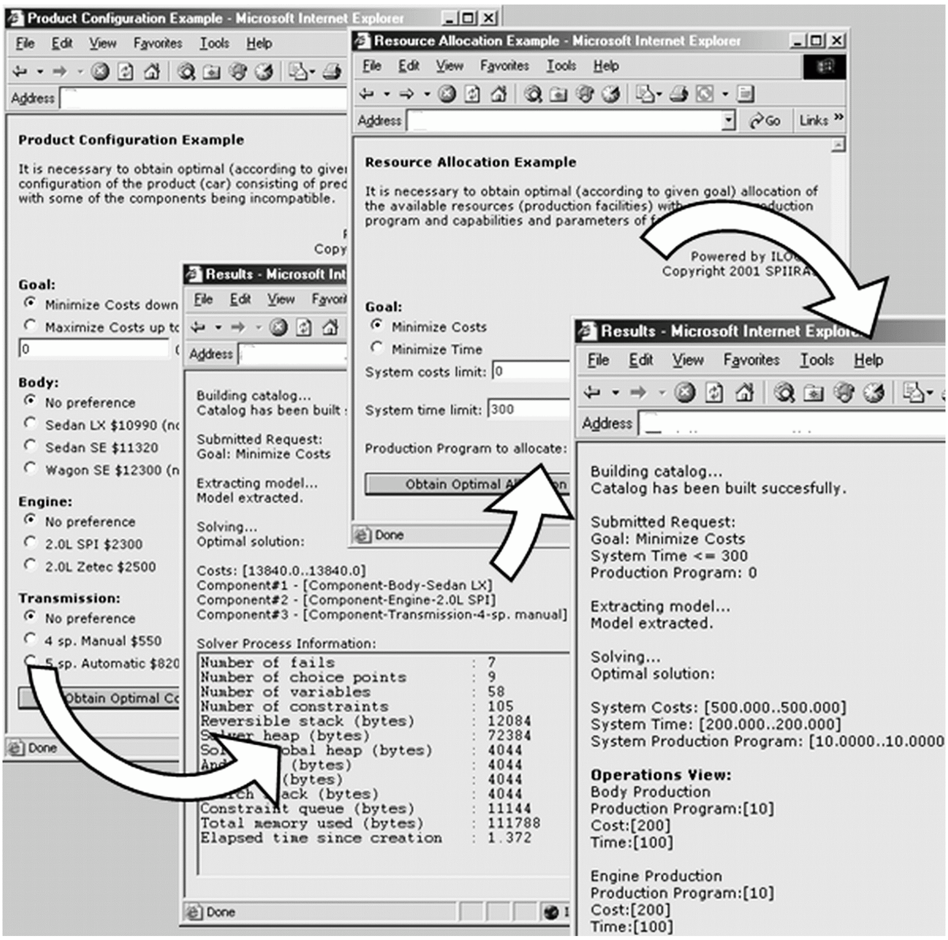


Fig. 26. User request input and output forms for car configuration and resource allocation examples

Since the examples were implemented as web-based applications, they can also be considered as prototype of the Multi-component product e-configuration tool [58].

6 DISCUSSION AND FUTURE WORK

Comparison of the KSNet system with some other existing KF-oriented systems/projects is presented in Table 5. They are:

- KRAFT (Knowledge Reuse and Fusion/Transformation) — multi-agent system for integration of heterogeneous information systems. The main aim of this project is to enable sharing and reuse of constraints embedded in heterogeneous databases and knowledge systems. It has a hierarchy of shared ontologies for local resource ontology translation.
- InfoSleuth — multi-agent system for retrieving and processing information in a network of heterogeneous information sources.

Future work includes development of models, methods, agent architectures and object-oriented conceptual projects for direct knowledge entry by problem domain experts and for knowledge repository parallel development by distributed teams and examining the effectiveness of the proposed approach in more practical applications.

One of the future tasks is an implementation of virtual reality-based ontology engineering environment using VRML. This will increase its efficiency due to combination of modelled images with our natural 3D perception of the world. Figure 27 demonstrates a sequence of prototyped VRML-based screens for "in-depth" search from vehicle taxonomy to car structure.

It is necessary to investigate and develop the problem-oriented agents' negotiation/cooperation models and algorithms for KSNet.

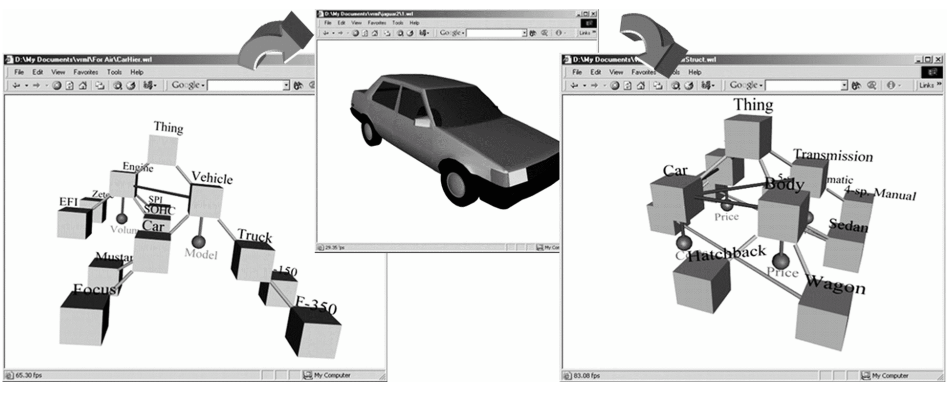


Fig. 27. Example VRML-based screenshots

Characteristic	KRAFT	InfoSleuth	KSNet
Languages and formats used	KQML, P/FDM, CoLan, CIF	KQML, KIF, OKBC, JDBC, LISP, CLISP, LDL+, Java, C/C++	KQML, KIF, DAML+OIL, MS Visual C++, ILOG, MS Access, HTML, JavaScript, PHP
Supported sources	Any available information sources for which appropriate processing mechanisms exist	Initially databases; currently any available sources for which appropriate processing mechanisms exist	Any available sources for which appropriate processing mechanisms exist
Multi-agent architecture	FIPA-based with peer-to-peer interaction	FIPA-based with mediating interaction	FIPA-based with mixed peer-to-peer & mediating interaction
Relationships between ontologies	Hierarchy	Mapping of sources ontologies to the system ontology.	Mapping of sources ontologies onto current application ontology.
Peculiarities	Processes data and constraints	The network of interacting agents is developed. Mechanisms of messages interchange in multi-agent systems are described	Utilizes object-oriented constraint networks for knowledge representation
Case study	Virtual enterprises	Environmental Date Exchange Network (EDEN) project	E-business, virtual enterprises

Table 5. Comparison of the KSNet system with existing knowledge/information integration systems

7 CONCLUSIONS

The paper discusses techniques, supporting procedures/tasks used for implementation of the knowledge fusion KSNet systems based on the KSNet-approach to knowledge logistics. The description of multi-agent architecture of the KSNet system based on this approach is given. The structure and major features of software prototype are presented. Given examples prove applicability of the developed techniques to such areas as management, product configuration, and supply chain. Consequently, this approach could be useful for such fields as e-business, configuration management, strategic planning, etc. Utilizing ontologies and compatibility with modern standards (such as DAML+OIL) allows seamless integration of the developed approach into existing processes in the described areas. The components of the system repository enable utilizing heterogeneous KSs due to application of top-level ontology, provide scalability due to expandable/renewable KB, and allow rapid knowledge search due to application of knowledge map and user profiles.

Acknowledgments

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

REFERENCES

[1] DAML — the DARPA Agent Markup Language web site. Available on: `www.daml.org`, 2001.

[2] DAML+OIL Web Site. Available on: `http://www.daml.org/2001/03/daml+oil-index.html`, 2001.

[3] Smirnov, A.—Pashkin, M.—Chilov, N.—Levashova, T.—Haritatos, F.: Knowledge Source Network Configuration Approach to Knowledge Logistics. International Journal of General Systems, Taylor & Francis Group, Vol. 32, No. 3, 2003, pp. 251–269.

[4] Weiss, G.: Introduction. In: G. Weiss (Eds.): Multiagent Systems: a Modern Approach to Distributed Artificial Intelligence. The MIT Press, Cambridge, Massachusetts, London, Vol. 8, 2000.

[5] Franklin, S.: Is It an Agent, or Just a Program?: A Taxonomy for Autonomous Agents. Proceedings of the Third International workshop on Agent Theories, Architectures, and Languages. Springer-Verlag 1996.

[6] Huhns, M. N.—Stephens, L. M.: Multiagent Systems and Societies of Agents. In: G. Weiss (Eds.): Multiagent Systems: a Modern Approach to Distributed Artificial Intelligence. The MIT Press, Cambridge, Massachusetts, London, 2000, pp. 79–120.

[7] Jennings, N. R.: On Agent-based Software Engineering. Artificial Intelligence, Vol. 117, 2000, pp. 277–296.

[8] Chandrasekaran, B.—Josephson, J. R.—Benjamins, V. R.: What are Ontologies, and Why Do We Need Them? IEEE Intelligent Systems & Their Applications, January–February 1999, pp. 20–26.

[9] Smirnov, A.—Chandra, C.: Ontology-Based Knowledge Management for Co-Operative Supply Chain Configuration. Proceedings of the 2000 AAAI Spring Sym-

posium "Bringing Knowledge to Business Processes". Stanford, California: AAAI Press, 2000, pp. 85–92.

[10] Rasmus, D. W.: Knowledge Management: More Than AI But Less Without It. PC AI, Vol. 14, 2000, No. 2, pp. 35–44.

[11] Jarrar, Y.—Schiuma, G.—Zairi, M.: Defining Organisational Knowledge: A Best Practice Perspective. VI International Conference on "Quality Innovation Knowledge", Kuala Lumpur, 17–20 February, 2002, pp. 486–496.

[12] Dzbor, M.—Paralic, J.—Paralic, M.: Knowledge Management in a Distributed Organisation: Proceedings of the BASYS'2000 — 4th IEEE/IFIP International Conference on Information Technology for Balanced Automation Systems in Manufacturing Kluwer Academic Publishers, London, September 2000, ISBN 0-7923-7958-6, pp. 339–348.

[13] Park, J. Y.—Gennari, J. H.—Musen, M. A.: Mappings for Reuse in Knowledge-based Systems. SMI Technical Report 97-0697, 1997.

[14] Ceruti, M.—Anken, C.—Lin, A.—Rubin, S.: Applications of High-Performance Knowledge-Based Technology. Proceedings of IEEE Systems Man and Cybernetics, Vol. 3, Nashville, Tennessee, USA, October 8–11, 2000.

[15] Vail, E. F.: Knowledge Mapping: Getting Started with Knowledge Management. Information Systems Management, Fall 1999, pp. 16–23.

[16] Microsoft Sharepoint Web Site. Available on: `http://www.microsoft.com/sharepoint/`, 2002.

[17] Hummingbird Web Site. Available on: `http://www.hummingbird.com/products/eip/`, 2002.

[18] IBM Lotus Discovery Server web site. Available on: `http://www.lotus.com/home.nsf/welcome/discoveryserver`, 2002.

[19] Maedche, A.—Staab, S.: Discovering Conceptual Relations from Text. In: W. Horn (Ed.): Proceedings of the 14th European Conference on Artificial Intelligence (ECAI 2000), IOS Press, Amsterdam 2000.

[20] Tecuci, G.—Boicu, M.—Marcu, D. et al.: Developpment of a Disciple Agent for Center of Gravity Analysis. Proceedings of the 18th National Conference on Artificial Intelligence and the 14th Conference on Innovative Applications of Artificial Intelligence, Edmonton, Alberta, Canada, July 27–August 1, 2002, pp. 853–860.

[21] Blythe, J.—Kim, J.—Ramachandran, S.—Gil Y.: An Integrated Environment for Knowledge Acquisition. Proceedings of the International Conference on Intelligent User Interfaces, Santa Fe, New Mexico, January 2001, pp. 13–20.

[22] Trellis Project Web Site. Available on: `http://www.isi.edu/expect/projects/trellis/index.html`, 2002.

[23] Cogito: E-Commerce with Guaiding Agents Based on Personalised Interaction Tools. Available on: `http://ipsi.fhg.de/~cogito`, 2002.

[24] Cheah, Y.—Abidi, S. S. R.: Healthcare Knowledge Acquisition: An Ontology-Based Approach Using the Extensible Markup Language (XML). Available on: `http://www.geocities.com/yncheah/papers/mie00_cheah_abidi_ont.pdf`, 2002

[25] Sure, Y.: A Tool-supported Methodology for Ontology-based Knowledge Management. In: E. Stubkjaer (Ed.): The Ontology and Modeling of Real Estate Trans-

actions in European Juristictions. International Land Management Series, Ashgate, Available on: `http://www.informatik.uni-bremen.de/ heiner/COST/Sure.doc`, 2002.

[26] Ontoprise: Semantics for the WEB. Available on: `http://www.ontoprise.de/products/index_html_en`, 2003.

[27] Protégé-2000 Project web site. USA, Stanford Medical Informatics at the Stanford University School of Medicine. Available on: `http://protege.stanford.edu/index.shtml`, 2000

[28] Ontolingua web site. Stanford University, Knowledge Systems Laboratory. Available on: `http://www-ksl-svc.stanford.edu:5915/&service=frame-editor`, 2001.

[29] Pease, A.—Chaudhri, V.—Lehmann, F.—Farquhar, A.: Practical Knowledge Representation and the DARPA High Performance Knowledge Bases Project. In A. Cohn, F. Giunchiglia, B. Selman (Eds.): Proceedings of the Conference on Knowledge Representation and Reasoning. KR-2000, Breckenridge, CO, USA, Morgan Kaufmann, April 2000, pp. 12–15.

[30] AKT project web site. Available on: `http://www.aktors.org/`.

[31] Visser, P. R. S.—Jones, D. M.—Beer, M. D.—Bench-Capon, T. J. M.—Diaz, B. M.—Shave, M. J. R.: Resolving Ontological Heterogeneity in the KRAFT project. Proceeding of the International Conference On Database and Expert System Applications, DEXA-99, Florence, Italy, Springer-Verlag LNCS, 1999.

[32] Nodine, M. H.—Unruh, A.: Facilitating Open Communicating in Agent Systems: the InfoSleuth Infrastructure. Technical Report MCC-INSL-056-97, Microelectronics and Computer Technology Corporation, Austin, Texas 1997, 78759.

[33] Mena, E.—Kashyap, V.—Sheth, A.—Illarramendi, A.: Estimating Information Loss for Multi-ontology Based Query Processing. Proceedings of the 13th Biennial European Conference on Artificial Intelligence, ECAI'98, Workshop on Intelligent Information Integration, III98, Brighton, UK 1998.

[34] Smirnov, A.—Pashkin, M.—Chilov, N.—Levashova, T.: Multi-agent Architecture for Knowledge Fusion from Distributed Sources. Lecture Notes in Artificial Intelligence. Springer, 2002, 2296, pp. 293–302.

[35] Adams, M. B.—Deutsch, O. L.—Hall, W. D. et al.: Closed-Loop Operation of Large-Scale Enterprise: Application of a Decomposition Approach. Proceedings of DARPA ISO Symposium on Advanced in Enterprise Control (AEC), San-Diego, USA, 2000, pp. 1–10.

[36] Howells, H.—Davies, A.—Macauley, B.—Zancanato, R.: Large Scale Knowledge Based Systems for Airborne Decision Support. In: Knowledge-Based Systems, Vol. 12, pp. 215–222.

[37] DARPA Advanced Logistics Project web site. Available on: `www.darpa.mil/iso/alp`, 2002.

[38] Cunningham, M.: Buyer (and Seller). Be Aware. In: DB2 Magazine, Vol. 6, No. 2, pp. 33–37.

[39] Kim, J.—Ling, R.—WILL, P.: Ontology Engineering for Active Catalog.: Proceedings of AAAI Workshop on Using AI in Electronic Commerce, Virtual Organizations and Enterprise Knowledge Management to Reengineer the Corpora-

tion, Providence, Rhode Island. Available on: `http://www.isi.edu/expect/link/papers-ontos.html`.

[40] Kurbel, K.—Loutchko, I. A.: Framework for Multi-Aagent Electronic Marketplaces: Analysis and Classification of Existing Systems. Proceedings of the International NAISO Congress on Information Science Innovations, ISI'2001, Symposium on Intelligent Automated Manufacturing, IAM'2001, Dubai, U. A. E. Electronic Proceedings, 2001.

[41] Anken, C. S.: Information Understanding. Introduction for 5th Anniversary Information Workshop, Rome, NY. The Information Institute, Air Force Research Laboratory (AFRL) Information Directorate, 2002.

[42] Varshney, P. K.: Scanning the Special Issue on Data Fusion. Proceedings of the IEEE, Vol. 85, 1997, pp. 3–5.

[43] Holsapple, C. V.—Singh, M.: The Knowledge Chain Model: Activities for Competitiveness. Expert System with Applications, Vol. 20, 2001, pp. 77–98.

[44] Aguirre, J. L.—Brena, R.—Cantu, F. J.: Multiagent-based Knowledge Networks. Expert Systems with Applications 2001, Vol. 20, pp. 65–75.

[45] Gray, P. M. D.—Hui, K.—Preece, A. D.: Finding and Moving Constraints in Cyberspace. AAAI-99 Spring Symposium on Intelligent Agents in Cyberspace, Stanford University, California, USA. AAAI Press 1999, pp. 121–127.

[46] Smirnov, A.—Pashkin, M.—Chilov, N.—Levashova, T.: Multi-Agent Knowledge Logistics System "KSNet": Implementation and Case Study for Coalition Operations. In: V. Marik, J. Müller, M. Pechoucek (Eds.): Multi-Agent Systems and Applications. Proceedings of the 3rd International Central and Eastern European Conference on Multi-Agent Systems, CEEMAS 2003, Prague, Czech Republic, June 16–18, 2003. Proceedings LNAI 2691, pp. 292 ff.

[47] Smirnov, A.—Pashkin, M.—Chilov, N.: Agent Architecture and Negotiation Protocol for Knowledge Logistics in Distributed Intelligent Enterprise. In: A. Dolgui, J. Sofdek, O. Zaikin (Eds.): Production System Design, Supply Chain Management and Logistic. Proceedings of the 9th International Multi-Conference on Advanced Computer Systems (ACS'2002), Szczecin, Poland 2002, Part 2, pp. 379–386.

[48] Smirnov, A.—Pashkin, M.—Chilov, N.—Levashova, T.: Multi-agent Support of Mass Customization for Corporate Knowledge Management. Proceedings of the International Conference on Intelligent Manufacturing Systems, 2003, Budapest, Hungary, pp. 103–108.

[49] FIPA Documentation. Geneva, Switzerland, Foundation for Intelligent Physical Agents (FIPA). Available on: `http://www.fipa.org`, 2000.

[50] Guan, S.—Zhu, F.—Ko, C. C.: Agent Fabrication and Authorization in Agent-Based Electronic Commerce. Proceedings the International ICSC Symposium on Multi-Agents and Mobile Agents in Virtual Organizations and E-Commerce, MAMA'2000, Wollongong, Australia, Electronic Proceedings, 2000.

[51] Duecker, M.—Thies, S.: A Framework for Knowledge Acquisition from Multiple Experts. Proceedings of the International NAISO Congress on Information Science Innovations (ISI'2001), Symposium on Intelligent Automated Manufacturing, IAM'2001, Dubai, U. A. E. Electronic Proceedings, 2001.

[52] Smirnov, A.—Pashkin, M.—Rakhmanova, I.: Multi-agent WWW-based Environment for Distributed Team Decision Making Support. Proceedings of Benefit Day on Pan-European Co-operation and Technology Transfer, Budapest, Hungary 1997, pp. 23–29.

[53] Rios, S.: Decision Theory and Decision Analysis: Trends and Challenges. Kluwer Academic Publisher, Boston Hardbound, ISBN 0-7923-9466-6 July 1994, 312 pp.

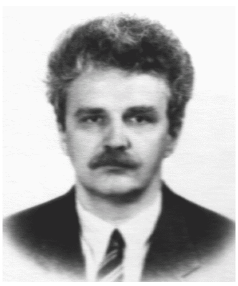
[54] Bach, E.—Glaser, E.—Condon, A.—Tanguay, C.: DNA Models and Algorithms for NP-complete Problems. Proceedings of the 11th Annual IEEE Conference on Computational Complexity, CCC'96, Philadelphia, Pennsylvania, USA 1996, pp. 290–300.

[55] ILOG Corporate web site. Available on: `http://www.ilog.com`, 2001.

[56] Chaudhri, V. K.—Lowrance, J. D.—Stickel, M. E.—Thomere, J. F.—WADLINGER, R. J.: Ontology Construction Toolkit. Technical Note Ontology, AI Center. Report, January 2000, SRI Project No. 1633, 85 pp.

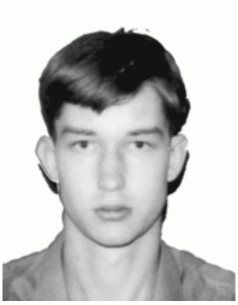
[57] Maccregor, R.—Patil, R. S.: Tools for Assembling and Managing Scalable Knowledge Bases. Available on: `http://www.isi.edu/isd/OntoLoom/hpkb/OntoLoom.html`, 1997.

[58] Akkermans, H.: OBELIX Project Proposal: The EU e-Business Ontology Project. Vrije Universiteit. Available on: `http://www.ontoweb.org/workshop/ucecOntoWeb/obelix.ppt`.



Alexander V. Smirnov, Prof., received his ME, PhD and D.Sc. degrees in St. Petersburg, Russia, in 1979, 1984, and 1994 respectively. He is the Deputy Director for Research and a 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 Polytechnical University and St. Petersburg State Electrical Engineering University. His current research interests include corporate knowledge management, multi-agent systems, group decision support systems, virtual enterprises, and supply chain management. He has published more than 150 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 a researcher at 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 a researcher at the Computer Aided Integrated Systems Laboratory of SPIIRAS. His current research interests belong to areas of virtual enterprise configuration, supply chain management, knowledge management, ontology engineering and multi-agent systems. He is the author/co-author of more than 30 research papers published in proceedings of international conferences and books.

Tatiana V. Levashova received her ME degree from St. Petersburg State Electrical Engineering University in 1986. She is a leader programmer at Computer Aided Integrated Systems Laboratory of SPIIRAS. Her current research covers 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.