Knowledge Sharing for Continuous Business Engineering Based on Web Intelligence

A.Smirnov, M.Pashkin, N.Chilov, T.Levashova & A.Krizhanovsky
St.Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences
St.Petersburg, Russia

ABSTRACT: Modern business trends require from companies continuous business engineering to guarantee a continuous business improvements. Since the Internet has become an easy accessible and popular place business applications the problem of knowledge sharing based on using Web tools and dealing with knowledge representation and processing becomes actual. Existing tools are usually oriented to good structured knowledge sources processing and do not take into account features of model of users' interests. The paper presents a multi-agent architecture of the systems addressing knowledge logistics as a tool providing for a web-based intelligent service of continuous business engineering.

1 INTRODUCTION

The major trend of knowledge-dominated economy is a shift from "capital-intensive business environment" to "intelligence-intensive business environment". This shift requires global changes in business paradigms. While the traditional thinking about the business involves people, processes, and technologies, the modern business systems will consider two more factors: infrastructure and strategy. In case of the business infrastructure the major goal of Enterprise Application Integration have to go beyond sharing information to rapidly adapting to changing business conditions. It is required to wide implementation of continuous business engineering (CBE) ideas to industrial project practice.

CBE uses information and interpretation of factors from the target and external business infrastructure. It does not require the gathering or ownership of this information, as there are organisations specifically focuses in these areas. Information can be obtained as and when required.

From CBE perspective, the goal of EAI is sharing data and processes between an applications and information source in the enterprise. According to (Gormly, 2001) EAI involves integration at the following levels:

- Business Process Integration when an enterprise has to define, enable and manage the processes for the exchange of enterprise information among diverse business systems.
- Application Integration aimed at bringing data or a function from one application together with that of another application that together provide near real-time integration.
- Data Integration assuming that data must be identified (where it is located), catalogued, and a metadata model must be built (a master guide for various data stores). Once these three steps are finished, data can then be shared / distributed across database systems.
- Standards of Integration promoting the sharing and distribution of information and business data – standards that are at the core of Enterprise Application Integration including such standards as COM+/DCOM, CORBA, XML, etc.).
- Platform Integration assuming integration of the underlying architecture, software, hardware, and separate heterogeneous networks have to be integrated.

An efficient approach is required in order to provide a mechanism which would enable the above types of integration, what in turn would make possible for a business system to quickly react on changes in its environment and to be flexible enough.

Increases in the amount of required knowledge and intensive cooperation for CBE have caused a need to support an efficient knowledge sharing (import, capture, retrieval & access, and use of knowledge) and exchange between members of the business infrastructure so that the right knowledge from distributed sources can be integrated and transferred to the right person within the right context, at the right time, for the right purpose. The study of operations constituting these activities has led to the formulation of a new scientific area in knowledge management (KM) called knowledge logistics (KL) (Smirnov et al. 2002a). KL is based on individual user requirements, available knowledge sources, and current situation analysis. This technology enables EAI in the following aspects: business process integration, application integration, data integration.

Proposed in the paper KSNet-approach to knowledge logistics is oriented to “Just-Before-Time” service for intelligent support of CBE. Advanced technologies of intelligent agents, ontology management, constraint satisfaction, profiling, and knowledge fusion underlie the proposed approach.

Today, Web services are believed to be the crucial technology for business. Web service can be seen as high-level interfaces through which partners can conduct business operations. The priority modern industrial projects involve Web applications. This is not surprising because business is becoming more Internet-dependent. The integration of Web applications with existing systems is a key driver of EIA (Brown, 2003). Wide spread of modern information technologies, such as World Wide Web and intelligence agents (Huhns and Stephens, 2000), has led to an appearance of a new direction for scientific research and development called "Web intelligence".

Web intelligence explores fundamental and practical impacts of AI and advanced Information Technologies on the next generation of Web-empowered systems, services, and environments. The basis for the ongoing Web intelligence research agenda is made up the issues: (i) Web mining that applies data mining techniques to large Web data repositories; (ii) Web-based knowledge processing and management that focuses on developing the semantic Web, the base of this research is ontological knowledge representation; (iii) distributed inference engines that perform automatic reasoning on the Web; (iv) information exchange and knowledge sharing coupled with human-crafted resources that support sustainable knowledge creation (Zhong et al. 2002).

The paper is organized as follows. Section 2 elucidates knowledge logistics concerned with ontology approach. Section 3 presents main components of the system “KSNet”. Section 4 describes chosen knowledge sharing model of the system “KSNet”, architecture of the developed research prototype and web agent architecture. Section 5 presents Knowledge Fusion agent features as the most important problem-oriented web agent of the system. Main features of the system that correspond to fundamental capabilities of the intelligent Web’s design and development are presented in conclusion.

2 ONTOLOGY-DRIVEN KNOWLEDGE LOGISTICS

Knowledge logistics addresses the problem of acquisition of the right knowledge from distributed sources, its integration and transfer to the right person within the right context, at the right time, for the right purpose. This problem in the approach is considered as a network configuration that includes end-users, loosely coupled knowledge sources, and a set of tools and methods for knowledge processing located in an e-business environment. Such network of loosely coupled sources was referred to as the knowledge source network or “KSNet”.

The main principles considered during the development of the proposed approach and a KL system based on it originate from the characteristics of modern “e”-applications. These applications widely use ontologies as a common language for business process / enterprise modelling (Goossenaerts & Pelletier 2001, O’Leary 2000, OILEd 2002, Protégé 2003, Semantic Web 2003). Thus, the approach focuses on utilizing reusable knowledge through shared ontological representations.

The application of intelligent agents representing their knowledge via ontologies (Weiss, 2000) was motivated by the need of knowledge logistics systems for flexibility, scalability, and customizability. The multiagent system architecture based on the FIPA Reference Model (FIPA 2002) was chosen as a technological basis for the definition of agents’ properties and functions since it provides standards for heterogeneous interacting agents and agent-based systems, and specifies ontologies and negotiation protocols.

As a formal model for knowledge integration the ontology model with the knowledge representation formalism of object-oriented constraint networks was chosen. This allows simplifying the formulation and interpretation of real-world problems which in the areas of engineering, manufacturing, management, etc. are usually presented as constraint satisfaction problems (Smirnov et al. 2002a).

The object-oriented constraint networks formalism (Smirnov, 2001) was chosen as the abstract model for ontology representation (Fig. 1). The abstract model based on this notation unifies main concepts of languages, such as standard object-oriented languages with classes, and constraint programming languages. It supports the declarative representation, efficiency of dynamic constraint solving, and problem modelling capability, maintainability, reusability, and extensibility of the object-oriented technology.

According to the paradigm the knowledge can be described by classes, attributes, domains, constraints, and methods. This perspective of knowledge representation correlates well with the semantic metadata representation concept being developed under the Semantic Web project (Semantic Web, 2003).

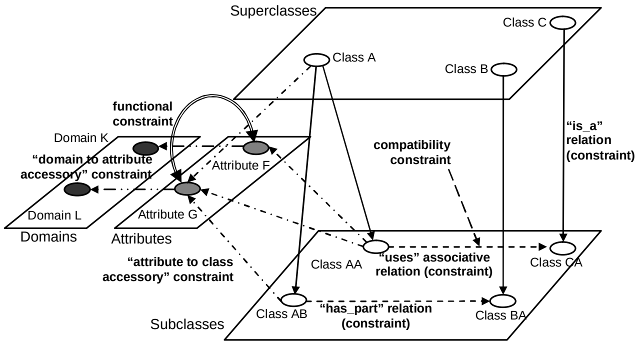

Figure 1. Object-oriented constraint network notation as knowledge representation formalism

3 SYSTEM "KSNET"

The framework of the proposed approach is implemented in the system "KSNet". The system architecture takes into account such modern requirements to applications as flexibility, learning from the user, integrity, velocity, open connectivity, reasoning, and customizability.

The system applies ontologies for user request processing. The following ontology types for the system were defined: (i) top-level ontology describing notation for ontology representation in the system; (ii) application ontology (AO) describing an application domain in terms of domain and tasks & methods ontologies; (iii) preliminary knowledge source (KS) ontology describing KS in KS's terms and the top-level ontology notation; (iv) KS ontology (KSO) containing correspondence between terms of KS and AO; (v) preliminary request ontology describing user request in user's terms (which are used by the user for requests input) and the top-level ontology notation, (vi) request ontology containing correspondence between terms of preliminary request ontology and AO; (vii) domain ontology representing static knowledge about a particular domain in terms of the domain; and (viii) tasks & methods ontology describing problem-solving knowledge in terms of a domain or high-level terms that are general for several domains. The ontologies are stored in a common ontology library (OL) that allows sharing and reusing them.

The OL's ontologies share a common notation of object-oriented constraint networks provided by the top-level ontology. Domain ontologies and tasks & methods ontologies are formed as a new knowledge becomes available. The new knowledge here is knowledge provided by experts, retrieved from KSs, or obtained as results of user request processing. Both new ontologies can be created (if there is no ontology relating to domain/task/method of the new knowledge) and existing ontologies can be expanded (otherwise).

Relationships between a domain ontology and a tasks & methods ontology are established if knowledge of the domain is used by a task or a method described by the tasks & methods ontology. According to the chosen formalism the tasks and methods of the tasks & methods ontology are described by ontology classes, output and input parameters of the tasks/methods are described by attributes of these classes. If an attribute value from a domain ontology can be considered as a parameter of a task/method then an associative relationship is established between the domain ontology class holding the attribute and the class of the tasks & methods ontology representing the task/method and vice versa.

Parts of domain ontologies and tasks & methods ontologies make up an AO. AO is a conceptual model describing a real-world application domain. It depends on a particular domain and task. In case, when a request has not been processed before, parts of domain and tasks & methods ontologies relevant to the request are integrated into a new AO. Otherwise, an existing AO is reused. Request ontology is formed by alignment of preliminary request ontology and AO that is to be used for the request processing. Alignment is defined as establishing links between two ontologies by the definition of a correspondence between their elements.

A conceptual scheme of the user-oriented ontology-driven KL methodology is presented in Figure 2. The system works in terms of a common ontology library's vocabulary. Each user / user group works in terms of an associated expandable request ontology and thereby with a part of AO corresponding to their requests and consequently to the requests ontologies. 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 which KSs are to be used for the request processing in the most efficient way. For this purpose the knowledge map including information about locations of KSs is used. Translation between the system's and KS' notations & terms is performed using KSOs.

During KSO creation (when a new KS is attached to the system) and modification (when an appropriate KS is changed), a correspondence between KS terms and the AO terms is identified. As a result of this process a set of corresponding KSs is defined for classes and their attributes from the application ontology. This set is stored in the knowledge map and used for preparation of a user-oriented KS network configuration by the configuration agent.

The system "KSNet" (Fig. 3) uses intelligent agents to provide access to distributed heterogeneous KSs. Multiagent systems offer an efficient way

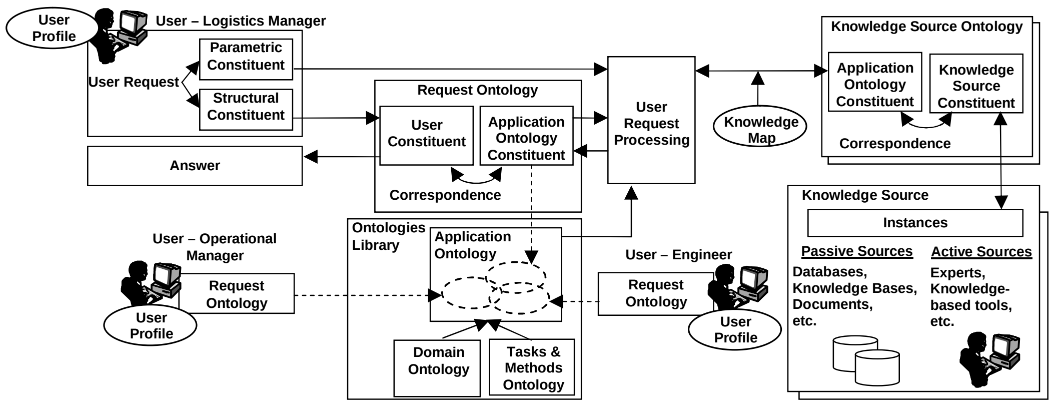

Figure 2. Conceptual scheme of the user-oriented ontology-driven knowledge logistics methodology

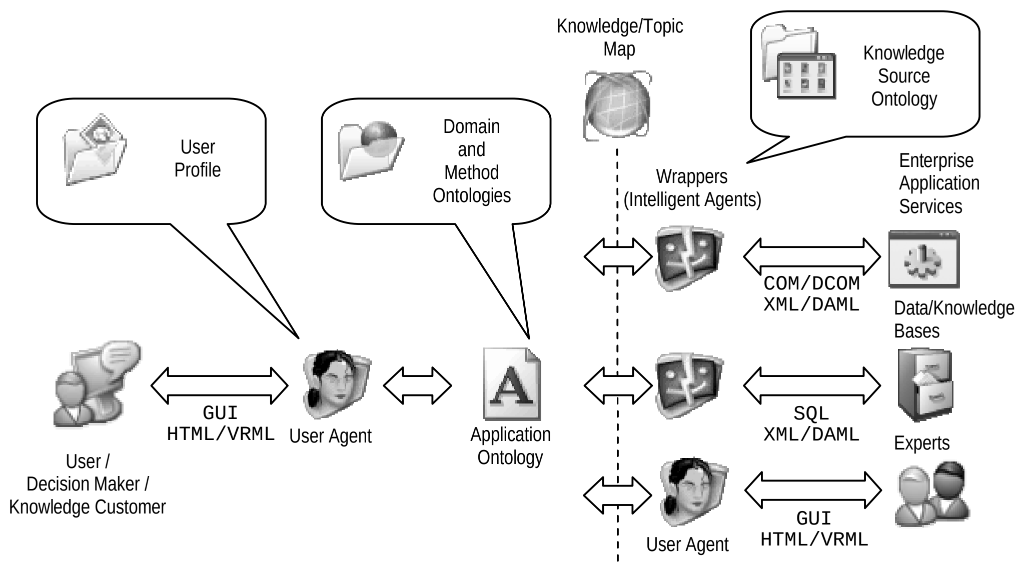

Figure 3. Generic architecture of the system "KSNet"

to understand, manage, and use the distributed, large-scale, dynamic, open, and heterogeneous computing and information systems (Weiss, 2000). 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 knowledge logistics tasks and scenarios for their collaboration were developed: translation agent (terms translation between different vocabularies), knowledge fusion agent (knowledge fusion 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). The multi-agent architecture is described in detail in (Smirnov et al. 2001, Smirnov et al. 2002b).

4 WEB-BASED KNOWLEDGE SHARING

One of the most important areas of the system "KSNet" is knowledge sharing. The scientific direction of knowledge sharing is devoted to design and development of methods, models and tools enabling sharing and reuse of distributed knowledge presented in various heterogeneous formats. It can be considered as a technology providing for means to such items of "Knowledge Process" as import, capture, retrieval & access, and use of knowledge. Knowledge is distinguished in two types: (i) explicit knowledge: knowledge that was acquired, validated, structured, and saved, and (ii) tacit knowledge: knowledge that can be acquired using special tools (at present it does not acquired).

As a basis for the implementation of agents' models, methods and functions for the knowledge sharing in the system "KSNet" the "tacit-explicit model" is used (Lindvall et al. 2002, Nonaka 1994, Nonaka & Takeuchi, 1995). This model describes four types of knowledge translation: *tacit—tacit* (transferring knowledge through sharing experiences, working together on a team, direct exchange of knowledge), *tacit—explicit* (transforming personal knowledge to espoused knowledge that can be either recorded or unrecorded), *explicit—tacit* (assimilating knowledge acquired from knowledge items by people) and *explicit—explicit* (a reconfiguration of explicit knowledge through sorting, adding, combining, and categorizing). Table 1 presents a list of the system agents and components providing for different types of knowledge translation. Basic model of knowledge sharing in the system "KSNet" is presented in Figure 4.

Table 1. Knowledge translation support in the system "KSNet"

Knowledge translation	Agents	Repository elements
Tacit-tacit	User agent Expert assistant agent	User profiles
Tacit—explicit	User agent Expert assistant agent Ontology management agent Monitoring Agent Mediator	User profiles Internal knowledge base Knowledge map Ontology library (application, domain, task & methods ontologies)
Explicit—tacit	User agent Translation agent Ontology management agent Monitoring Agent	User profiles Ontology library (application, request, task & methods ontologies)
Explicit—explicit	All the system agents	All the repository elements

5 KNOWLEDGE FUSION AGENT AS AN EXAMPLE OF CONTINUOUS BUSINESS ENGINEERING IMPLEMENTATION

The user request defines both the problem statement and what data has to be retrieved from OL and from KSs. Thereby the problem statement is changed from one request to another. The novel "on-the-fly" compilation mechanism in combination with ILOG (ILOG, 2003) (a generic tool for object-oriented constraint programming) was proposed to solve these varying problems. This mechanism assumes that the Knowledge Fusion (KF) agent gathers required data from some other KSNet agents (translation agent, ontology management, configuration agent and wrapper), generates a solver (performs "on-the-fly" compilation), and launches the solver to generate a solution set.

The KF agent performs knowledge fusion based on AO, user RO, and knowledge acquired from KSs. The implementation of KF agent uses such fundamental ideas of programming languages as object-oriented languages using classes and constraint programming languages. The ILOG Configurator (ILOG, 2003) was chosen as a generic tool for object-oriented constraint programming. It provides a library of re-usable and maintainable C++ classes. Those classes define objects in the application domain in a natural and intuitively way so that it is possible cleanly distinguish the problem ***representation*** from the problem ***resolution***. Therefore if a problem statement has changed then it is not necessary to rewrite the entire code as in case of "pure" C++. In the given case the problem statement is defined by data retrieved from the OL, therefore the problem of minimal code modification, fast and error-free is the important task here.

The novel on-the-fly compilation mechanism was proposed to solve this problem. The essence of this mechanism is to write the AO's task definition (classes, attributes, constraints) to a C++ file directly. AO is based on domain, tasks & methods ontologies stored in the OL. Elements of AO are obtained from the OL. Thus the KF agent creates C++ file based on these data and copies it to a program (Microsoft Visual Studio project) prepared in advance. The program is compiled in order to create an executable file in the form of dynamic-link library (DLL). After that the KF agent calls the function from DLL to solve the task. The UML sequence diagram (Figure 5.) shows the KF agent at the stages of knowledge obtaining, solver compilation and execution.

The task to be solved by the KF agent is to generate a solution set (if it exists) using the AO and ontology created on basis of the user request. The KF agent uses the mentioned technique of dynamic code generation (with ILOG Configurator commands embedded) to produce a solution set satisfying requirements of the user request as well as AO rules (classes relations, constraints on attributes, etc.).

The generated code (C++ file) is based on the data obtained from KS. It consists of several parts:

- The ontology management agent creates a part of the program based on data from the OL;
- The wrapper creates a part of the program using local/remote KSs;
- The KF agent generates a part of the program based on user request processing as well as user requirements and interests;

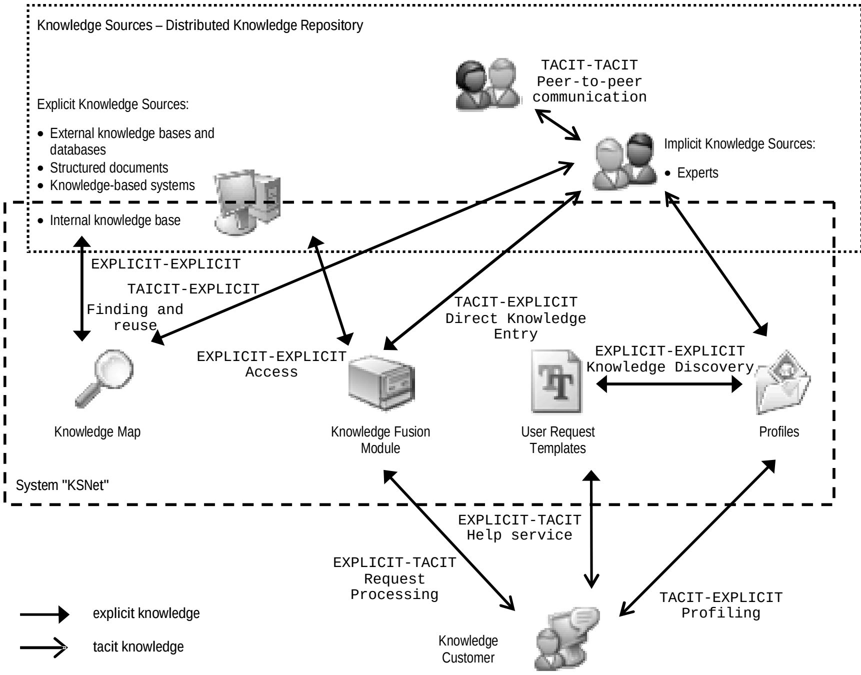

Figure 4. Basic model of knowledge sharing in the system “KSNet”

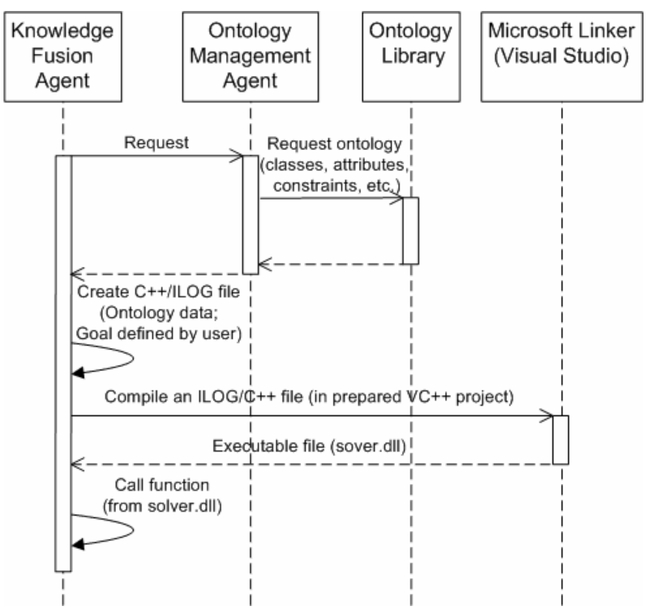

Figure 5. The UML sequence diagram shows the KF agent scenario

– The predefined part of code (unchangeable): an algorithm and strategy definition and an automatic answer generation.

6 CONCLUSION

Among the advantages of the presented system “KSNet” the following regarding to Web intelligence requirements were selected: (i) agents’ properties and functions enable to organize the agent community in a way to support autonomic Web; (ii) ontology representation paradigm facilitates to process and understand natural language; (iii) ontology library based on the common vocabulary and notation can be considered as a dynamically created source of metaknowledge (iv) user profiles and request ontologies support the personalization requirement; (v) translation of ontologies from advanced formats (e.g., DAML+OIL) into internal representation and out of it enables knowledge sharing and reuse; (vi) knowledge map plays a role of a distributed knowledge repository; (iii) the technology of constraint satisfaction & propagation enables to perform automatic reasoning on the Web.

Web-agent based architecture is a good basis for web intelligence support of continuous business engineering. It allows to minimize requirements to user computers and allows users to have only HTML-compatible Web-browser and access to the Internet. Intelligent web-agents can act in a distributed environment, independently from the users and apply ontologies for knowledge representation and interchange. Developed agents’ functions for the current situation monitoring allow for agents to rapidly execute appropriate scenarios.

Using the knowledge fusion agent based on the novel mechanism of the “on-the-fly” compilation allows generating new knowledge that is not available in existing knowledge sources independently on the application domain, and content & context of the current problem.

7 ACKNOWLEDGEMENTS

Some parts of the research were done under the partner project ISTC # 1993P funded by US Air Force Research Laboratory at Rome, NY, the project # 2.44 of the research program "Mathematical Modelling, Intelligent Systems and Nonlinear Mechanical Systems Control" of the Presidium of the Russian Academy of Sciences, the grant # 02-01-00284 of the Russian Foundation for Basic Research, and the project of the research program "Fundamental Basics of Information Technologies and Computer Systems" of the Russian Academy of Sciences.

8 REFERENCES

Brown, T.M. 2003. The Integration Report. *eAI Journal* 2: 23—25.

Gormly, L. 2001. EAI Overview, URL: http://eai.ittoolbox.com/pub/eai_overview.htm

Goossenaerts, J. & Pelletier, C. 2001. Enterprise Ontologies and Knowledge Management. *Proceedings of the 7th International Conference on Concurrent Enterprising "Engineering the Knowledge Economy through Co-operation"*, pages 281—285, Bremen, Germany.

Huhns, M.N. & Stephens, L.M. 2000. Multiagent Systems and Societies of Agents. *In: Weiss, G. (ed.): Multiagent systems: a Modern Approach to Distributed Artificial Intelligence*. The MIT Press, Cambridge, Massachusetts, London: 79—120.

Lindvall, M., Rus, I., Sinha, S.S. 2002. Technology Support for Knowledge Management. *In: Proceedings of the 4th International Workshop on Learning Software Organizations (LSO '02).* Chicago, Illinois, USA, URL: http://csce.unl.edu/LSO2002/Lindvall-Rus-Sinha.pdf.

Marik, V., Pechoucek, M. & Bárta, J. 2002. Reduction of Task Complexity in Coalition Planning. *Proceedings of the International Symposium "From Agent Theory to Agent Implementation" (AT2AI-3),* URL: http://www.ai.univie.ac.at/%7Epaolo/conf/at2ai3/-final/at2ai3Marik.pdf, Vienna, Austria (EU).

Nonaka, I. 1994. A Dynamic Theory of Organizational Knowledge Creation. *In: Organization Science*, 5(1): 14—37

Nonaka, I. & Takeuchi. H. 1995. *The Knowledge Creating Company*. Oxford University Press.

O'Leary, D.E. 2000. Different Firms, Different Ontologies, and No One Best Ontology. *IEEE Intelligent Systems*, 15(5): 72—78.

Rathmell, R.A. 1999. A Coalition Force Scenario "Binni – Gateway to the Golden Bowl of Africa". *Proceedings on the International Workshop on Knowledge-Based Planning for Coalition Forces*: 115—125, Edinburgh, Scotland.

Smirnov, A., Pashkin, M., Chilov, N. & Levashova, T. 2002. Multi-agent Architecture for Knowledge Fusion from Distributed Sources. *Lecture Notes in Artificial Intelligence*, (2296): 293—302, Springer.

Smirnov, A. 2001. Ontology Drive Virtual Production Network Configuration: a Concept and Constraint-Object-Oriented Knowledge Management. *Proceedings of the Third International Conference on Enterprise Information Systems (ICEIS 2001)*: 345—352, Setubal, Portugal.

Smirnov, A., Pashkin, M., Chilov, N., Levashova, T. & Haritatos, F. 2002a. A KSNet-Approach to Knowledge Logistics in Distributed Environment. *Proceedings of the International Conference on Human-Computer Interaction in Aeronautics (HCI-Aero 2002)*. pages 88—93, USA, Menlo Park, AAAI Press.

Smirnov, A., Pashkin, M., Chilov, N. & Levashova, T. 2002b. Agent-Based Knowledge Fusion in Scalable Information Environment: Major Principles and System Framework. *Proceedings of the First International ICSC Congress on Autonomous Intelligent Systems (ICAIS'2002)*, ISBN: 3-906454-30-4, Geelong, Australia.

Weiss, G. 2000. *Multiagent Systems: a Modern Approach to Distributed Artificial Intelligence*. The MIT Press, Cambridge, Massachusetts, London.

Ning Zhong, Jiming Liu, & Yiyu Yao. In Search of the Wisdom Web. *Computer* 35 (11): 27—31.

9 ADDITIONAL REFERENCES

Clin-Act, 2000 Clinical Activity 2000. The ON9.3 Library of Ontologies: Ontology Group of IP-CNR, 2002, URL: http://saussure.irmkant.rm.cnr.it/onto/.

Cyc, 1998 Hpkb-Upper-Level-Kernel-Latest: Upper Cyc/HPKB IKB Ontology with links to SENSUS, Version 1.4, 1998. Ontolingua Ontology Server, 2002, URL: http://www-ksl-svc.stanford.edu:5915.

FIPA 2002 Foundation for Intelligent Physical Agents Documentation, 2002, URL: http://www.fipa.org.

ILOG, 2003 ILOG Corporate Web-site, 2003, URL: http://www.ilog.com.

Loom, 1997 Loom ontology browser 1997. Information sciences Institute, The University of Southern California, 2002, URL: http://sevak.isi.edu:4676/loom/shuttle.html.

NAICS, 2001 North American Industry Classification System code, DAML Ontology Library, Stanford University, URL: http://opencyc.sourceforge.net/daml/naics.daml.

OILEd, 2002 OILEd web site, URL: http://img.cs.man.ac.uk/-oil/.

Protégé, 2003 Protégé Project, USA, Stanford Medical Informatics at the Stanford University School of Medicine, 2003, URL: http://protege.stanford.edu/.

Semantic Web, 2003 The Semantic Web Community Portal, URL: http://www.semanticweb.org.

UNSPSC, 2001 Universal Standard Products and Services Classification Code, DAML Ontology Library, Stanford University, URL: http://www.ksl.stanford.edu/projects/-DAML/UNSPSC.daml.