AN AGENT-BASED APPROACH TO THE DEVELOPMENT OF INFORMATION SYSTEMS FOR MILITARY LOGISTICS

Stanimir STOJANOV, Roumen VENKOV and Radi RADEV

1. Introduction

The development of modern information systems in various application areas (e-commerce, logistics, and intelligent telecommunication services) presents a serious challenge to software technologies. This can be acknowledged to the difficulty, complexity, heterogeneity and distributed functionality of the systems. It is still a difficult task to find a homogeneous technological framework for the effective solution of these problems. Different models and standard frameworks have been developed for this kind of systems.[1] Although they posses certain advantages, the requirement for generality makes their application in specific domains relatively expensive. In addition, considerable resources are needed for their adaptation.[2] Agent-based systems solve some of these problems, especially those systems developed in e-commerce applications.[3] The agent-based technology offers flexibility and easier adaptation.

To provide efficient development facilities, modern software technologies have to meet different, often contradictory, requirements. The point here is that they have to strike a balance between user requirements and required technological support. From this point of view, the technology is expected to effectively solve the following two problems:

- To support the execution of the distributed functionality of the systems, and
- To provide the user with easy and intuitive interface for work with the system, while the internal complexity and heterogeneity of the system remains hidden.

INFORMATION & SECURITY. An International Journal, Vol. 8, No. 2, 2002, 239-251.

The approach presented in this paper is characterized by the following two phases:

1. Definition of a model that meets the specific requirements and reflects the organization of the military logistics in the Bulgarian Army;
2. Use of the agent-based framework MALINA for the implementation of a prototype based on the model.

Following these two stages, we intend to improve the system performance.

2. A Model of the Military Logistics Support

The following general considerations form the basis for development of a model for military logistics. From a user's perspective, the distributed information system could be considered as an architecture consisting of two parts: a server and a client. The server provides a given functionality accessed trough requests to the system. The processes performed are transparent to the users. The client part is the interface between users and the system. In order to facilitate the interaction, it is necessary to provide a powerful technological support, comprised of a multi-layered distributed architecture, application software (implemented as objects, components or autonomous agents), standardized internal interfaces, and relevant control mechanisms.

2.1. Services

In the adopted approach the concept *service* is a basic notion. From one point of view, the service is a specific interface which users use for interaction with the system. Furthermore, using this concept, we intend to strike a balance between user and technological aspects of a complex distributed system.

From a user's perspective the service is a logically separated functionality and understanding and implementing it does not create problems. However, it is not an easy task to define the services from a technological point of view. Adapting the service architecture to conform to the requirements of our approach, involves difficulties such as flexibility, transparency and adaptability of the program components. Moreover, the components have to operate in a dynamic environment and they are under the control of different types of management mechanisms (centralized or decentralized). In this context, we define the services as generic structures, consisting of two components:

- Kernel, that realizes the functionality (business-logic) of the service;
- Shell, containing all necessary software modules that allow the execution of the service in an operational environment.

The services can be implemented as objects, components or agents. The specification of standardized, logically separated functionality depends on the particular application area. For instance, in e-commerce the standardized functions are specified by Electronic Commerce Building Blocks (ECBB), [4] while in telecommunication Service Independent building Blocks (SIBs) are used.[5,6]

2.2. Model overview

Generally, the systems for military logistics are rather complex. They are built on heterogeneous platforms, and often it is necessary to distribute their functionality. To deal with the complexity associated with military logistics, we have decomposed the problem. The elaborated relevant and stable architecture and the distributed functionality are specified and presented on the logical model below (refer to Figure 1).

The model consists of three independent *horizontal levels*. These levels specify the static functionality of the system and reflect various degrees of generalization. The information systems of the individual units function on the *low level*. The *middle level* is formed by the information systems of the formations and the corps. The *high level* includes the systems of the Armed Forces and the General Staff.

Each level is built by a separate logistical chain. A chain integrates the activities, which are logically interrelated and applied to homogeneous objects. The following four types of logistical chains are defined in the model:

- *Material Chains* (MC) – information activities that organize, provide and control the movement of materials;
- *Financial Chains* (FC) – activities for management and control of the financial flows;
- *Personnel Resources* (PR) – information activity related to human resources management;
- *The Document Chains (DC)* – activities for organization and management of the documents.

The document processing chains are considered at a more general level. They are specific logistical chains with two functions. First, they are used for horizontal integration of the other three chains. Second, they have their own contents and specifics and they can be integrated vertically between the levels.

Within a given level, we allow integration of the logistical chains (we shall name it as *horizontal* integration). The horizontal integration can be automated and realized by an adequately integrated information system. Horizontal integration can be achieved between different logistical chains.

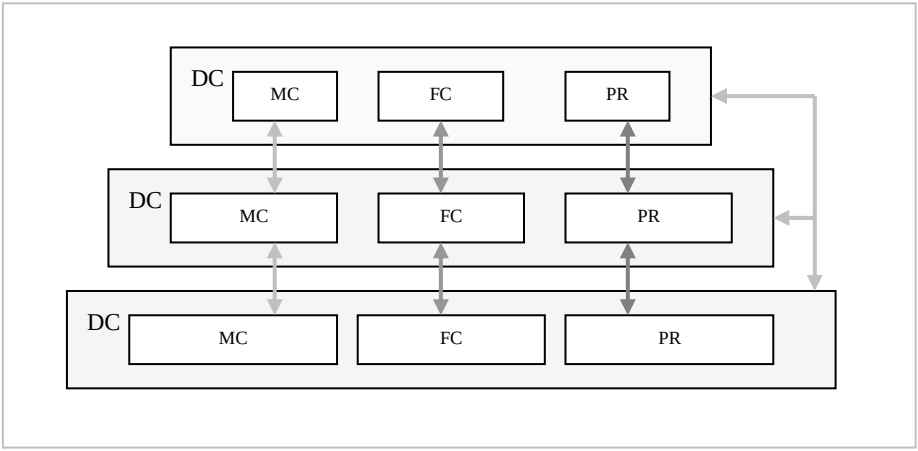


Figure 1

Between the different levels we also allow for the so-called *vertical* integration. The vertical integration is possible only between homogeneous logistical chains (chains of one type). This type of integration can be automated and serviced by the information systems of a corresponding application area. The *application area* is defined as a collection of the related homogeneous logistical chains of the different levels. The logistical chains of the different levels are integrated by means of standardized *interfaces.*

In the model, the operation and control of the programming modules is organized as *business-transactions*. It is possible to define separate standard business-transactions, which could be parameterized, customized, and modified. All this provides a satisfactory level of flexibility in the system. The business-transactions can be *horizontal* (within the framework of a technological chain or an integrated system) or *vertical* (within the framework of one or more applications). Horizontal business-transactions are, for example, the instance management and control of the *Food Supply Service*, depot facilities, the orders and deliveries, the financial services in the units. Vertical business-transactions are the management and control of the orders and deliveries in the Bulgarian Army, the management of the depot (warehouse) facilities at formation and corps level.

3. MALINA Technology

For development of the system we have applied a multi-agent technology known under the name MALINA (Multi-Agent Local Integrated Network Associations). A

thorough description of the technology can be found in various publications.[7,8,9,10] The technology supports a bottom-up approach to the development of multi-agent applications.[11] The MALINA technology specifies three levels in the development process:

- *Abstract level* – a hypothetical infrastructure, which provides the theoretical framework of the technology, is defined at this level.
- *Conceptual level* – concepts are decomposition and specification of the hypothetical infrastructure; they aid the development of supporting programming tools and an appropriate development environment. The following four concepts form the basis for establishing the development environment of the technology:
- *Abstract Agent Architecture* – it specifies the architecture of an abstract generic agent, by which the agents in different applications can be created;
- *AgentCities* – it specifies a static infrastructure for multi-agent applications;
- *AgentAssociations* – it provides various possibilities for building agents associations;
- *MobileServices* – it provides different modes for automatic generation of mobile agents;
- *Development level* – it is composed of the supporting tools and the development environment of the adopted technology.

3.1. Abstract Agent

This concept has two goals: to specify a basic architecture, which can be used as a generic framework for the creation of application agents and to define the agents' life cycle. The concept defines an architecture, which will be known as *Abstract Agent*, by means of which the agents in a particular application can be created. The Abstract Agent (see Figure 2) is a generic abstract structure $A = <\pi,\phi,\mu>$, where:

- π - denotes the *agent engine*, which includes the agent's interfaces to the environment (sensors and effectors) and the agent's local control;
- ϕ - is the *specialization* of the agent. This is an abstract interface to some software modules that implement the agent's functionality
- μ - represents the *mentality* of the agent. This in turn specifies also an abstract interface to a well structured knowledge base. The knowledge base consists of independent segments. Each segment includes knowledge and processing methods that define the agent's behavior, for example beliefs, intentions, and commitments.

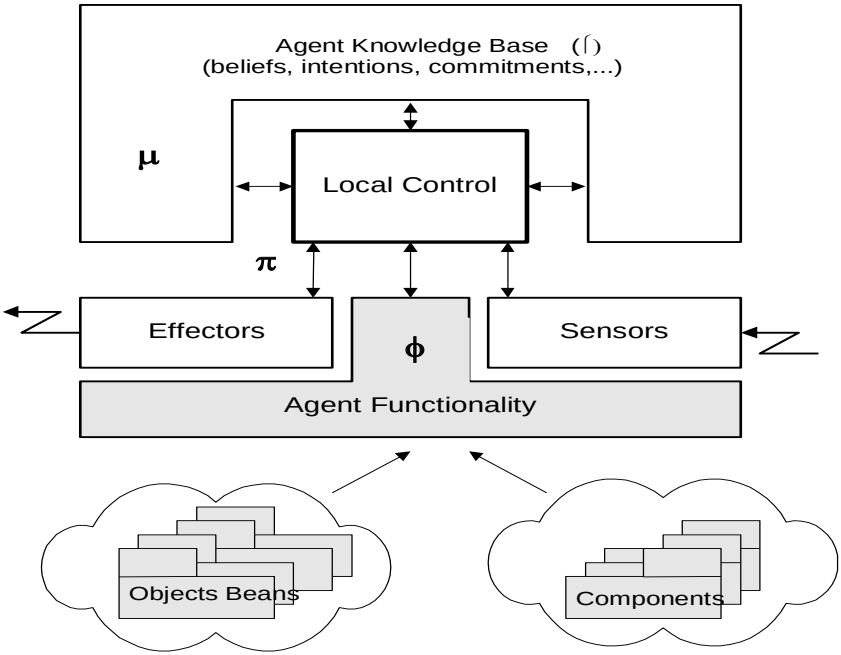


Figure 2

The Agent's Life Cycle is an abstract procedure that includes the following steps:Observe an input at time T

- Record any relevant information
- Check input consistency
- Solve the problem or generate a plan
- Compose answer expression
- Send the answer ***Agent Cities***

The concept *Agent Cities* specifies an open infrastructure, by which the multi-agent system can be modeled. The basic building block of the infrastructure is the "*city*". The "cities" are local virtual areas where the agents of the system will be located. The infrastructure is static, i.e. the "cities" have constant locations (addresses). The agents' locations within the "cities" are also static. In order to identify and control the "cities" and the agents, the concept defines and supports two address spaces – a logical address space and a physical address space.

Within the logical space the "cities" are identified by names (identifiers), which are unique in the system. The logical address of agents is described by a pair *(Name, Social role)*. The agent names (identifiers) are unique within the "cities". The logical address space is visible to users and system designers/developers. The agents in an application interact with their logical addresses.

The physical address space is used to support and control the messaging processes on a computer network. This address space is visible only during the run-time of a multi-agent application. To users and designers it is transparent.

3.3. Agent Associations

In conformity with the technology a multi-agent system is the set of all specified "cities". On the other hand, the set of the located agents in a "city" does not constitute an agent association. The interactions between the agents in a "city" are a source of additional semantics that cannot be defined only by the agents themselves. An agent association is specified further by the type of agents' interaction and by the organizational and management models of the "cities". In this concept, we distinguish three types of interactions between agents:

- *Communication* – the simplest possible interaction form, which supports the message exchange between the agents;
- *Synchronization* – a more complex interaction form, which can be used for synchronization of the agents. Synchronization is necessary for solving joint tasks. This type of interaction is applicable only between agents that show intelligent behavior (the so-called intelligent agents);
- *Cooperation* – the most complex form of interaction. It is used for preparation and planning of common work. It is applicable only for intelligent agents.

In order to implement the interaction concept we use the agent communication language KQML.[12] The technology supports two major types of control mechanisms in the "cities": centralized and decentralized management models. In the case of *centralized management* only a single intelligent agent is present in the "city" and it plays the role of a manager. The rest of the agents (the so-called workers) do not exhibit intelligence and they are subordinate to the manager. The manager receives the requests, distributes the work, and determines the responsibilities of the workers (i.e. it can be viewed as a kind of planner agent). In the case of d*ecentralized management* there are more intelligent agents present in the "city". In order to accomplish the requirements of a received request, the agents have to cooperate.

4. DIANA – A Multi-Agent Application in Military Logistics

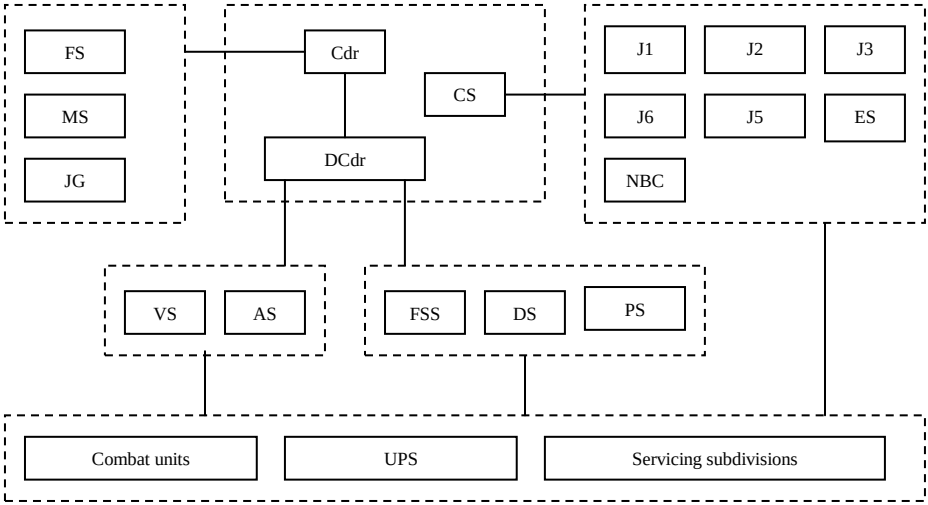
DIANA is a prototype of the information system for logistics support that implements the low level of the model described above. During system development, the model presented in Figure2, has been adapted to the organizational scheme of the real logistical system of the units. The system is designed as a three-layered architecture. The *first* layer is the client interface to the system, which enables the users (i.e., commanders, deputies of the commanders, the chiefs of the directorates and services, the warehousemen (the chiefs of the depots), the clerks, the suppliers (mess-sergeants), and others) to interact with the system. It controls the correctness of the incorporated information, generates the request and sends it to the application part of the system. The *second* layer is divided into two sub-layers – system and application. The system sub-layer controls and checks the processing of the different business transactions. It interprets the received order, generates the corresponding business transaction and controls its execution. This sub-layer supports also the horizontal integration of the system. The application layer contains separate program components, which realize the business logic of the different logistical chains (Figure 3). The database contains necessary basic nomenclatures.

The module "*Management and Control of the Food Supply Service*" has been implemented in the first version of DIANA. The module is designed as a multi-agent application with three groups of agents: system agents (SA), application agents (AA) and personal agents (PA). The SAs and AAs are server-based agents, i.e. they are transparent to the users. The business logic of the module is distributed among the functional kernels (the specialization) of the application agents. Some of the application agents are listed below:

- AA^{MG} – the agent is a generator of the menu-ration. In addition, it justifies and optimizes the created menu-ration;
- AA^{DM} – this agent controls the document processing in the system
- AA^{RG} – the agent is responsible for preparation of different reports
- AA^{WM} – a warehouse (food depot) manager
- AA^{DB} - this is a set of filter agents, which provide the access to the database.

The system agents assist in the operation of the application agents, where their primary task is to support the communication between the agents, the addressing of the agents and the management of the agent associations (cities). The users do not have a direct access to the system and to its information resources. The relations between the user and the server part of the system are realized by personal agents. They are visualized and they play different social roles (according to the terminology of MALINA), such as *Commander*, *Chief of the Food Supply Service*, *Clerks*, *Chief*

of the Food Depot. With the help of the development tools (Figure 4) of MALINA we can configure also other PAs or customize the SAs and AAs in accordance with the requirements of any new application.



LEGEND:
Cdr(Commander)
FS – Finance Section
DS – Dress Service
CS (Chief of Staff)
DCdr (Deputy Commander for Logistics)
MS (Medical Service)
PS (Petroleum, oil and lubricants Service)
JG (Jurisconsult General)
VS (Vehicle Service)
AS (Weapon Service)
ES (Engineer Section)
FSS (Food Supply Section)
NBC (NBC Service)
UCP(Units for Combat Provision)

Directorates:
J1(Personnel)
J2(Intelligence)
J3(Operative)
J5(Plans and Policy)
J6 (Communications-Electronics)

Figure 3

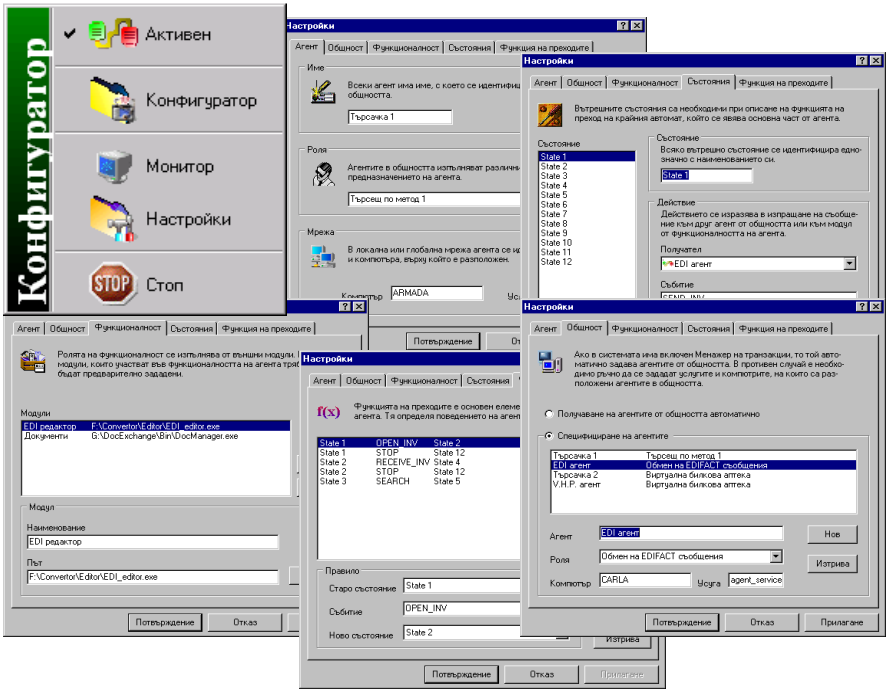


Figure 4

The DIANA-agents are deployed in the four “cities” (in the terminology of MALINA) described below, which form the infrastructure of the module. It is possible to make different arrangements. Figure 5 shows an infrastructure, which have been used during the testing of the system:

- C_{FSM} - Food Service Management
- C_{Cdr} – Commander (that only controls and signs documents)
- C_{WM} - Warehouse Management
- C_{DBS} – Database Services and Document Processing Services

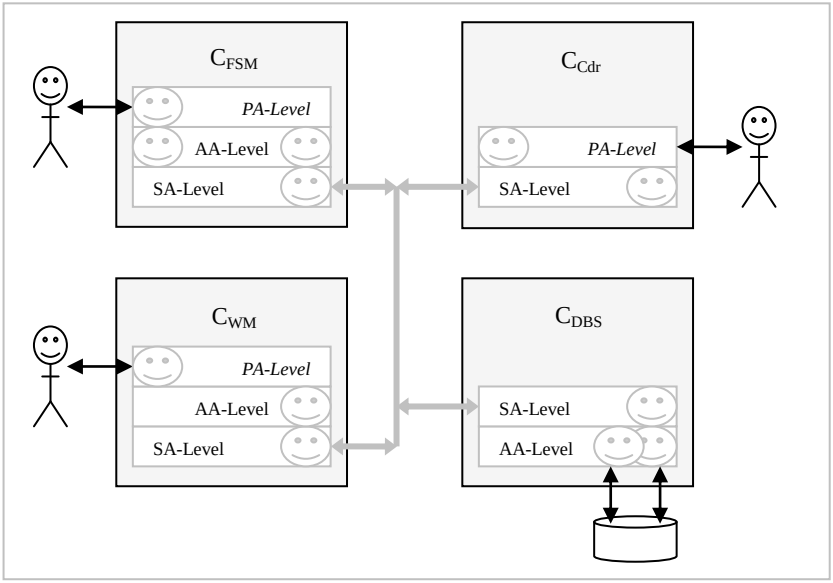


Figure 5

Figure 6 illustrates the administration module, by which we can create and support the information sources in the system.

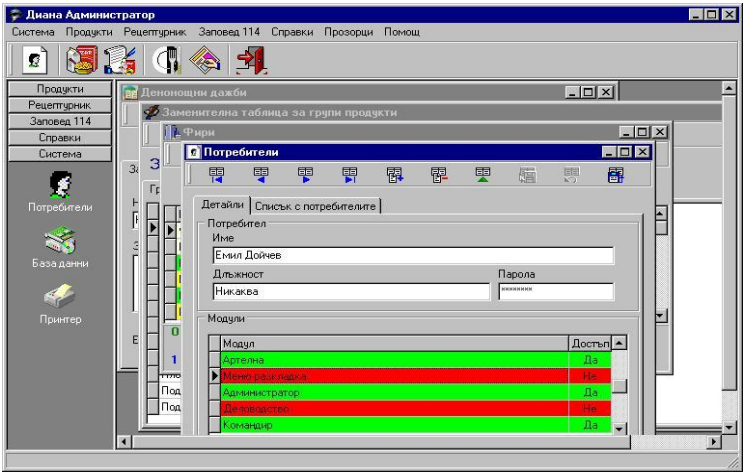


Figure 6

5. Conclusions

DIANA – the proposed system for military logistics – is implemented by means of the development tools of the MALINA technology on a J2EE platform and SQL Server. Experiments are performed using a system prototype in the unit 34560 in Yambol and in the Navy Base of Bourgas. The results of these initial experiments with the prototype demonstrate that the agent-based methodology is a viable approach to the development of military information systems.

Notes:

1 IBM, "San Francisco Product-Information," <http://www.ibm.com/Java/Sanfrancisco>, (1998); OMG, "CORBA BOCA – Business Object Component Architecture," *OMG-Document Nr. Bom/98-01-07*, (1998).

2 M. Merz, *Electronic Commerce*, (Dpunkt.verlag, 1999).

3 *BargainFinder*, <http://bf.cstar.ac.com/bf>; *Firefly*, <http://www.firefly.com>; *Jango*, <http://www.jango.com>; *Kasbah*, <http://kasbah.media.mit.edu>; *PersonaLogic*, <http://www.personalogic.com>; *Tete-a-tete*, http://ecommerce.media.mit.edu/tete-a-Tete (June 2002).

4 EBES/EWOS Project Team on Building Blocks for Electronic Commerce (PRB), "Building Blocks for Electronic Commerce," *Final Report*, (Brussels, 1997).

5 Q. Kong, G. Chen, and N. Lawler, "Distributed Architecture for IN Service Management," *IEEE Intelligent Network Workshop IN'96,*(The Carlton Crest Hotel, Melbourne, Australia, April 21-24, 1996).

6 SIEMENS, "IN Introduction, Reconfigured Services," (Vienna: IN Training Center, 1997); SIEMENS, "IN Introduction, System Overview and Functions," (Vienna: IN Training Center, 1997).

7 C. Wille, R. Dumke, and S. Stojanov, "Performance Engineering in Agent-Based Systems – Concepts, Modeling and Examples," in *Current Trends in Software Measurement, Proceedings of the 11th International Workshop on Software Measurement*, (Montreal, Canada, August 28-29, 2001), pp. 153-181.

8 S. Stojanov, M. Kumurdjieva, "MASTT - Technology and Its Application in Electronic Commerce Systems," in *Proceedings of the Workshop- Concurrency, Specification & Programming*, (Berlin: Humboldt-University, September 28-30, 1998).

9 S. Stojanov, M. Kumurdjieva, E. Dimitrov, and A. Schmietendorf, "Technological Framework for Development of Agent-based Applications," in *Proceedings of the Workshop - Concurrency, Specification & Programming*, (Berlin: Humboldt-University, October 9-11, 2000), pp. 299-311.

[10] S. Stojanov, I. Ganchev, "A multi-agent technology for creating of information systems for electronic commerce," in *Proceedings of the Second South Eastern European Conference on e-commerce*, (Sofia, 24-26 October 2000), pp. 92-96.

[11] H.D. Burkhard, "Theoretische Grundlagen in der verteilten Kuenstlichen Intelligenz," in *Verteilte kuenstliche Intelligenz*, ed. J. Mueller, (Germany: BI-Mannheim, 1993), pp. 157-189.

[12] T. Finin, J.Weber, G.Wiederhold, M.Genesereth, "Specification of the KQML Agent-Communication Language" *The DARPA Knowledge Sharing Initiative External Interfaces Working Group*, (University of Toronto, 1994); Y. Labrou, T.Finin, "A Proposal for a new KQML Specification", (University of Maryland Baltimore County, 1997).

STANIMIR STOJANOV is the head of the E-Commerce Laboratory at the University of Plovdiv. His scientific interests are in the fields of multi-agent technologies and systems, software architectures, intelligent telecommunication services, and E-Commerce. He is author of more than 50 papers and textbooks. *Address for correspondence:* Faculty of Mathematics and Informatics, University of Plovdiv, Plovdiv, Bulgaria; Phone: (++359 32) 277 292; E-mail: stani@uni-plovdiv.bg,

ROUMEN VENKOV is the general manager of ISY Intellect Ltd. His interests are in the fields of project management, configuration management, VL databases, and artificial intelligence. He is author of more than 30 papers and studies. *Address for correspondence:* 12 N.Gogol Street, Sofia, Bulgaria; Phone: (++359 2) 943 36 44; E-mail: isyint@bul.evro.net.

Colonel **RADI RADEV** is the ex-commander of the unit 34560 in Yambol where the model of the Military Logistician Support DIANA was realized. *Address for correspondence:* Druzhba-2, bl. 317, entr. 2, ap. 1, Sofia, Bulgaria, Phone: (++359 2) 983 83108.