Decision Making System for Operative Tasks

G. Shakah[1)], V.V. Krasnoproshin[2)], A.N. Valvachev[2)]
1) Irbid National University, Dept. of Computer Sci. Jordan, e-mail: minsk321@mail.ru
2) Belarus State University, Nezavisimost av. 4, 220050 Minsk, Belarus,
e-mail: krasnoproshin@bsu.by, van_955@mail.ru

Abstract: *Actual problems of construction of computer systems for operative tasks of decision making are considered. possibilities of solving the problems on the basis of the theory of active systems (tas) are investigated.*

Keywords**:** Decision making, active systems, distributed knowledge, multiagent systems

1.INTRODUCTION

Globalization has set a range of new medical, ecological, economic, social and other operative problems before the mankind. Their effective solution is possible on the basis of innovation knowledge irrespective of its location. "Innovation Knowledge" (IK) may be formalized information necessary for the solution to the problem. The information is characterized by topicality, accuracy and completeness. For effective use IK is represented in the form of computer Knowledge Bases (KB). Program systems, using KB, are known as Knowledge Based Systems (KBS) [1].

Nowadays the most topical are KBS for decision making or for refining (improving) of the made decision. The bases are usually expensive because they accumulate innovative expert knowledge with a possibility to replicate it for high quality problem solution in the distant parts of the earth. These very bases and systems are discussed in the paper.

Various models of knowledge representation (e.g. production rules, frames, semantic networks) and the corresponding

languages (e.g. Prolog, Lisp, ART, ILOG, OWL, CLIPS, KLM) are used for KBS construction. Generally, the existing knowledge bases have the following features: 1) locality; 2) integrity; 3) a small number of permanent knowledge sources; 4) complexity of modification; 5) mono-use; 6) high cost due to high percentage of manual work of the knowledge engineer and the programmer. KBS is usually developed by a local group with the use of a serial process of knowledge acquisition. This is acceptable for solving planned tasks the semantics of which is not being changed for years.

Evidently, technologies for solving operative tasks with rapidly changing semantics (on the basis of dynamically distributed knowledge sources characteristic for information and communication society) should form KB with opposite features: 1) state of distribution; 2) fragmentariness; 3) a great number of changeable sources; 4) simplicity of modification; 5) multi-use (including processing with Data Mining and Knowledge Discovery); 6) low cost due to automation of KB representation into the relevant computer system. These features make it possible to form KB by distributed groups within parallel processes of knowledge acquisition and ensure automatic generation of the target system. The existing technologies for KBS construction (Lotus Notes/Domino, Outlook/Projects/Exchange, Rational Rose, Rational XDE, Rational Requisite Pro, Rational Rapid Developer, Rational ClearCase) meet only some of the above mentioned features, they are expensive, complex in operation, much time is required for personnel training.

Thus, nowadays there is a need to develop a simple, inexpensive, intuitively understandable technology of operative KBS construction on the basis of geographically distributed knowledge sources excluding the role of the knowledge engineer and the programmer.

2. MAIN DEFINITIONS AND MODELS

Evolution of the initial problem statement into the target computer system for its solution is rather complex. Thus, the simplest methods should be used for the description of the process. One of them is an ontology approach realized in IDEF5 standard. A number of "rough" models of the problem are constructed. The models can be refined up to DLL process execution module. Within the approach let's specify the well-known AI definitions forming a comprehensive view of KBS development [2].

Global Information and Communication Medium (GICM) is an infrastructure, within which a problem is solved. GICM components are Internet, corporate networks, etc.

Task is a set of four constituents: problem statement (S), requirements to the solution (Req), solution result (Decision) and user manual (Manual):

$$\text{Task} = (S, \text{Req}, \text{Decision}, \text{Manual})$$

Group is a team participating in problem solution. *Group* consists of the center (Center), initiating the problem and distant executors (E) solving the problem:

$$\text{Group} = (\text{Center}, E^1, E^2, E^n)$$

Scene is a scheme including group participants, their roles, relations and means of information exchange (Com) within GICM:

$$\text{Scene} = (\text{Task}, \text{Group}, \text{Com})$$

Any modern problem is structured, thus it is necessary to decompose S into subproblems S^1, S^2, S^n. Operational efficiency of KBS development is closely related to time (T) and means (M) spending for the solution. Thus, let's refine the problem model:

$$\text{Task} = (S, S^1, S^2, S^n, \text{Req}^1,\ldots,\text{Req}^k, \text{Decision}^1,\ldots, \text{Decision}^n, \text{Manual}^1,\ldots,\text{Manual}^n, T^1,\ldots,T^n, M^1,\ldots,M^n), \text{ where: } n \rightarrow \infty, k = \text{variable}$$

Knowledge is structured and formalized information (Decision) necessary for the problem solution:

$$Decision = \sum Decision^i$$

Knowledge pattern is knowledge necessary and sufficient for subproblem solution including four constituents: problem statement, theoretical solution (dTheor), practical solution (dPrac) and the corresponding user manual:

$$Pat^i = Decision^i = (S^i, dTheor^i, dPrac^i, Manual^i)$$

Mediator is a portable program ensuring pattern transfer between the center and executors within GICM:

$$Mediator = (adrCenter, adrE^i, Pat^i)$$

where: adrCenter is center address, $AdrE^i$ is address of i-executor.

Knowledge domain (KD) is a set of patterns necessary and sufficient for the problem solution as a whole:

$$subPat = \sum Pat^i$$

Unenriched model of knowledge domain is KD model comprising structure of knowledge necessary for the problem solution.

Enriched model of knowledge domain is KD model comprising structure of the necessary knowledge as well as knowledge itself.

KBS is the result of KD representation into a computer system, the architecture of which includes subPat, navigation means (Menu), means for displaying subPat components (Content), programs of standard system service (Service), programs of application-specific processing (Proc) and built-in user manual:

KBS = (subPat, Menu=f '(subPat),
Content= f''(subPat,Menu), Service, Proc, Manual)

On the basis of the definitions we will formulate the general problem statement.

3. PROBLEM STATEMENT

Let there is a problem and a group for its solution. The center formulates problem S, decomposes it into subproblems S^1, S^2, S^n, distant executors E^1,E^2,E^n solve them in accordance with requirements $Req^1,\ldots,Req^k$. A set of decisions $Decision^1,\ldots, Decision^n$, represented in KBS form, is the result of the solution. It is necessary to develop a technology for transforming S into KBS providing time stabilization for the solution when the number of executors increases, i.e. $T(n) \approx T(m)$ when n=const, $m \to \infty$.

The solution of the general problem, according to modern approaches [1], requires the solution of the following subproblems:

- construction of the general scheme of the solution;
- construction of subPat KD model;
- development of process (P1) for forming KD model and its decomposition into fragments mPat;
- development of process (P2) for enriching model fragments by including subproblem solutions into mPat;
- development of process (P3) for integrating a set of mPat into uniform subPat KD;
- development of process (P4) for representing subPat into KBS.

Main requirements to the solution: 1) use of standard IBM-compatible PC; 2) use of OS Windows XP; 3) intuitively understandable interface.

4. SOLUTION SCHEME

Organization and communication scheme of KBS development is proposed as a basis for the technology construction. The scheme represents a standard process of computer system construction by distributed groups (Fig.1). The scheme is built on the basis of the theory of active systems [3] modified by the use of the above mentioned models of the center, executors and the mediator [2].

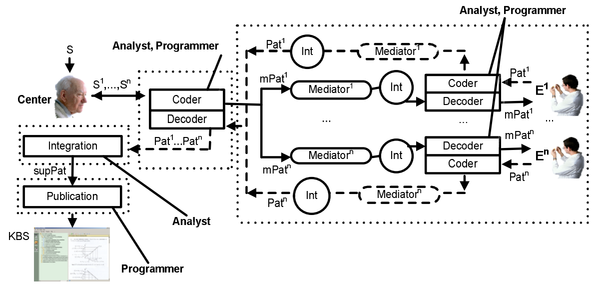

Fig. 1 – Organization and communication scheme of IT projects execution

According to the scheme, to automate transformation of S into KBS it is necessary to automate the activity of the analyst and the programmer delegating their functions to programs. The scheme is of distributed nature, thus it is appropriate to use a multi-agent approach [4]. But its realization, as shown in Fig.1, requires first of all the availability of knowledge domain model, a variant of which is described below.

5. KNOWLEDGE DOMAIN MODEL

We will use the well-known approach of Peter Chen [5] as the basis of KD model. The approach helps to structure semantic information about KD by extracting meaningful entities and relationship between them in the form of a hierarchy, known as ERD-model [2]. It is difficult to use it for distributed KD, because it is initially oriented to integral KD and local groups of executors.

To solve the problem of "KD state of distribution" we propose a dynamic model of mPat KD, comprising two parts (Fig.2) [2]. The static part is a graph, the vertex of which is problem S, nodes determine the hierarchy of subproblems $S^1,\ldots, S^n$, arcs represent the level of nesting. The dynamic part is represented by terminal vertices corresponding to Pat with added coordinates of the center (adrCenter) and the executor (adrE).

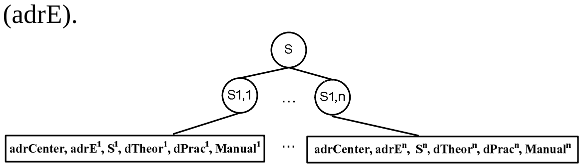

Fig. 2 – Dynamic model of mPat KD .

KD model, where $dTheor^i = \varnothing$, $dPrac^i = \varnothing$, $Manual^i = \varnothing$ is by definition unenriched. KD model, where $dTheor^i \neq \varnothing$, $dPrac^i \neq \varnothing$, $Manual^i \neq \varnothing$ is enriched. Thus, KBS development consists in realization of processes of KD model construction, enrichment of its dynamic components, their integration and representation into a computer system. The availability of a dynamic model of KD makes it possible to develop an architecture of agents realizing processes P1-P4 for its formation and use.

6. SYSTEM ARCHITECTURE

For realizing process P1 Modeller agent is proposed. The agent realizes three subprocesses: 1) construction of KD model in the center; 2) decomposition of the model into “unenriched” patterns; 3) delivery of “unenriched ” patterns to executors (Fig.3).

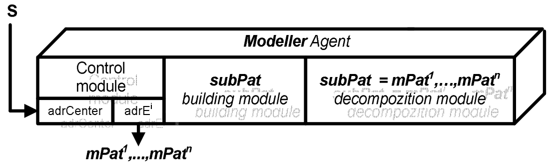

Fig. 3 - Architecture of Modeller agent

For realizing process P2 Miner agent is proposed. The agent realizes three subprocesses: 1) subproblem visualization; 2) inclusion of subproblem solution into the pattern; 3) sending of the pattern to the center (Fig. 4).

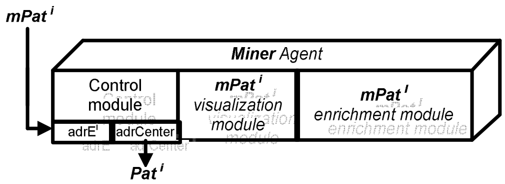

Fig. 4 – Architecture of Miner agent

For realizing process P3 Integrator agent is proposed. The agent realizes two subprocesses:

1) integration of patterns into KD; 2) analysis of KD completeness (Fig.5).

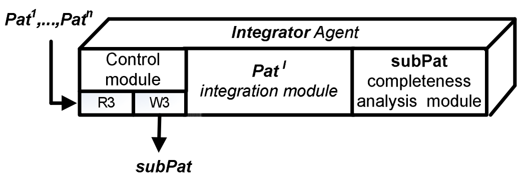

Fig 5 – Architecture of Integrator agent

For realizing process P4 Publicator agent is proposed. The agent realizes three sub processes:

1) construction of a tree for content management; 2) content selection; 3) assembling of the target system (Fig. 6).

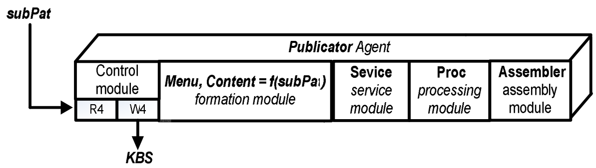

Fig. 6 – Architecture of Publicator agent

Assembler model (when forming interface) uses graphic jpg images, prepared in advance in Photoshop, the number of which is specified by the interface type.

In the traditional scene of KBS realization by replacing the analyst and programmer with the above described program agents, we will obtain a scheme of computer-assisted KBD development, where the only animate participant is Center (Fig. 7).

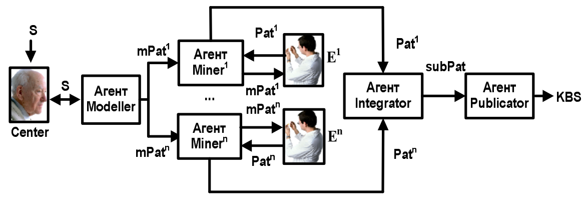

Fig. 7 – Multi-Agent Architecture

The architecture is realized in C# within .Net platform.

7. SYSTEM REALIZATION

A multilayer approach is used for realizing the architecture. Modeller, Integrator, Publicator agents are joined in the SDP Workshop program for the center (Fig.8a). Miner agent is integrated with Mediator SDP Expert program for executors (рис.8b).

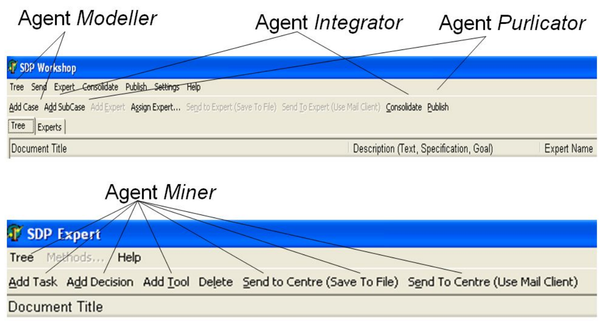

Fig. 8 – Interfaces of the center and executor: WorkShop (8a), SDP Expert (8b)

Two variants of KD representation are possible: network and local. In the first instance, KD is placed on the server with the help of standard means and browsed (MS Explorer, Opera, etc.) by users (Fig.9).

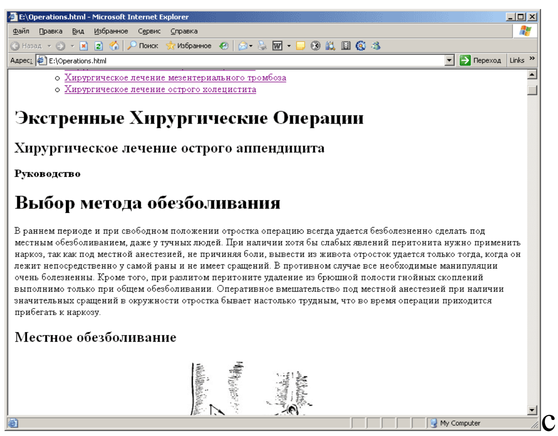

Fig. 9 – Variants of network interface

In the latter case, a system is formed on CD with the possibility to use on a computer that is not connected to Internet (Fig.10). At present a local KBS can have one of four possible variants of the interface.

The proposed multilayer architecture was used for realizing web-projects “Urgent Surgical Operations“ and “Delphi Programming” (fig.9), local projects “Orthopedic Cases“, “Urgent Surgical Operations“ (Fig.10), “Atlas of Forensic Medicine”, etc.

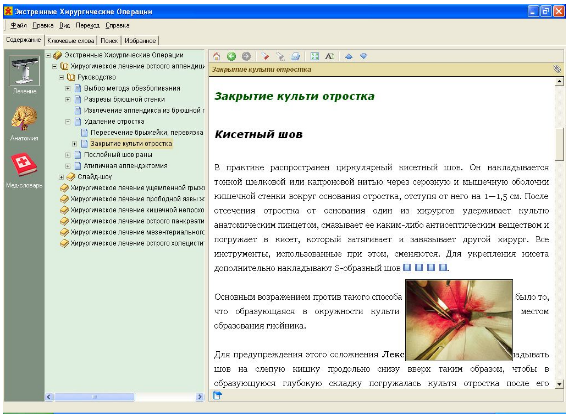

Fig.10 – Variants of local interface

Exclusion of the programmer role helped to solve the problem of execution time errors typical for KBS. The analyst did not participate either, because modeling and project decomposition were carried out by the center. From the user’s point of view the systems are of great help due to the multi-aspect application of KD.

In the large, the effectiveness of the developed technology is ensured by considerable reduction of time and means for KD construction and KBS formation due to the automation of the most labor-intensive processes.

8. CONCLUSION

The paper deals with the technology construction for operative development of systems based on knowledge.

A set of definitions, specifying (from the structural point of view) the well-known notions of AI with regard to KBS is proposed. A general scheme of the problem and project solution is formed, typical processes of realization are singled out. For each process a set of models describing the complete life cycle of KBS in Internet infrastructure has been developed. Intellectual functions of the analyst and the programmer have been realized in the form of agents on the basis of the multi-agent approach of the model. Agents are realized in C# for .Net within the architecture and the corresponding technology for automation of the most labor-intensive operations of KBS construction. Examples of the technology use for solving practical problems in medicine, programming and arts are presented.

The simplicity of the knowledge base modeling, automation of intellectual functions of the analyst and the programmer, parallel processes of acquisition of knowledge fragments, automation of their integration and publication resulted in considerable reduction of time for knowledge base construction by distributed executors. Intellectual activity of the analyst and the programmer is realized by taking into account model components specific for the knowledge model and introduction of the corresponding standards of knowledge representation. The module of knowledge domain processing can’t be standardized due to the specific nature of a particular project.

Development of the technology anticipates the increase of the number of possible variants of KBS interface and processing libraries included into the target system depending on the semantics of knowledge domains.

9. REFERENCES

[1] D.B.Lenat, R.V.Guha Building Large Knowledge-Based Systems. –Addison-Wesley,–391 p.

[2] V.V.Krasnoproshin Knowledge–Based Systems: Problems and Solutions // *Proceedings of the International Conference on Modeling and Simulation (MS’2006).* – Konya: Turkey, 2006. – Vol. 1. – P. 1–4.

[3] U.Boggs, M.Boggs *UML and Rational Rose*. – M.: Lori, 2000. – 580 c.

[4] G.Shakah, V.V.Krasnoproshin, A.N.Valvachev Fuzzy Approach to Active System Management. // *The Fifth International Conference on Neural Networks and Artificial Intelligence*. May 27-30, 2008, Minsk Belarus. – P.268-270.

[5] V.N.Burkov, D.A..Novikov T*heory of Active Systems: condition and prospects.* – M.: Sinteg, 1999. – 128 c.

[6] M.Wooldridge *Multiagent Systems*. – John Wiley & Sons. – 2002. – 340 p.