THE FRAMEWORK TO CREATE THE SYSTEM OF SELF-OPTIMIZING MOBILE AGENTS

Aleksej Otwagin[1)], Rauf Sadykhov[1)], Alexander Chelalo[2)]

[1)] Belarusian Academy of Sciences, United Institute of Informatics Problems (UIIP), 6 Sourganov str., Minsk, 220012, Belarus. e-mail: forlelik@yahoo.com.

[2)] Belarusian State University of Informatics and Radioelectronics, 6, P.Brovka str., Minsk, 220027, Belarus.

Abstract: Frameworks to design the modern multi-agent systems (MAS) are considered. Opportunities for introduction optimization algorithms in framework are offered. The architecture of the built-in system of MAS optimization during runtime on the basis of algorithm of a virtual network is considered.

1. Introduction

Modern multi-agent systems (MAS) find application in many areas of a science and technology. It is connected with fact, that for the solution of challenges, which demand the great computing performance, parallel computers are often applied. However, to develop the software for such systems an essential knowledge of their architecture and features are required.

On the one hand, the problem of software designing can be solved by introduction of design automation tools at various levels of the project procedure. These tools allow the developer to concentrate on algorithm of a task, and not paying attention to mechanisms of implementation of parallelism in application. However, tools for implementation of the parallel virtual machine cannot provide application mobility. Thus the developer should strictly adhere to the certain hardware architecture. As example various libraries of parallel programming should be considered.

On the other hand, the developer can use decomposition of algorithm on separate modules, each of which solves the part of the general problem. Modules are projected on some common programming language. Then modules will be made out as separate processes, which are capable to work within the framework of collective interaction support. The dialogue between modules should be fully transparent from the point of view of the developer.

The second approach is realized in modern MAS. In this systems the separate subtasks are “agents” that together try to find the solution, which satisfying some criteria. Specifically, agents should be planned the running and interaction so that to achieve the maximal benefit at the solution of the part of a problem. If high efficiency of functioning is achieved for all agents, then the whole system will also solve the problem more effectively.

As planning is very complicated task, there is a necessity to create the planning methods which allow to receive high efficiency of the overall problem solution at small expenses.

1. A survey of agent’s frameworks.

The agent, as described in FIPA specification [4], is the independent essence capable to reaction into environment, the definition of the targets corresponding to its interests, its actions in the world, and also deadlines and expenses for achievement of these targets. For effective development such independent, intellectual, reactive software parts, it is necessary to create good tools of implementation, debugging and estimation. For this purpose it is necessary to develop the framework allowing quickly to create agents of various types and to facilitate introduction of new technologies.

At framework construction stage the architecture that would isolate the logic of behavior dependent on the agent from the basic service code, which is common for all agents at modeling or application running, should be developed. Thus the new methodology to design the program systems, for example, component-based architectures. These architectures attempt to encapsulate the functionality of an object while respecting interface conventions, thereby enabling the creation of stand-alone applications by simply plugging together groups of components. This paradigm is ideal for agent framework, because it permits the creation of a number of common-use components, and other domain-dependent component can easily make use of in a plug-and-play manner.

The agent framework, created in conformity with the FIPA specification, has the architecture as shown on fig. 1 [4].

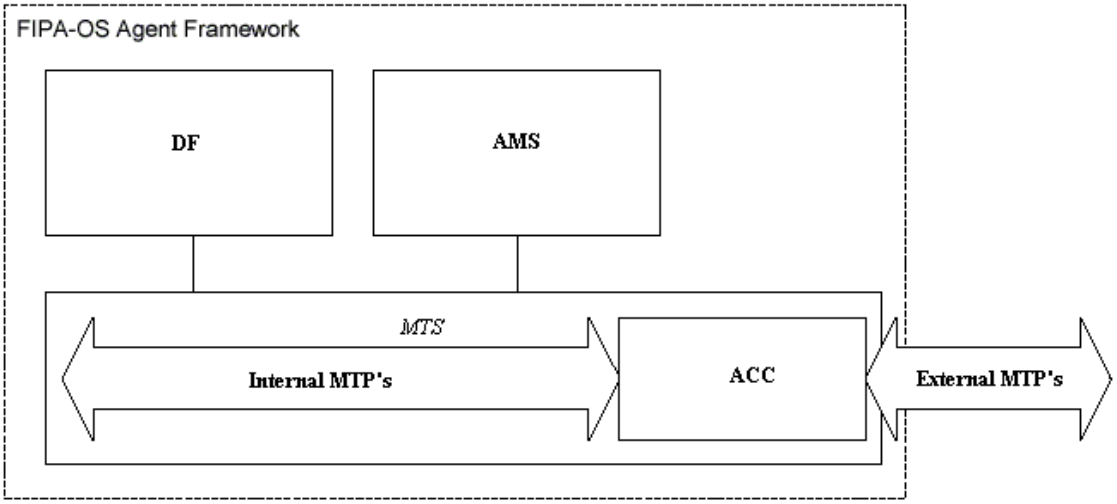

Fig. 1. Common architecture for agent's framework.

The Directory Facilitator (DF) and Agent Management System (AMS) are specific types of agents, which support agent management, and the Agent Communication Channel (ACC) is a lower-level entity that is part of the MTS (Message Transport Service). The DF provides "yellow pages" services to other agents. The AMS provides platform management functionality, such as monitoring the agent lifecycles and ensuring correct behavior of entities within the platform. The ACC supports interoperability both within and across different platforms; therefore, it is viewed as a component of the MTS. The MTS provides a message routing service for agents on a particular platform.

The AMS, MTS and DF form the Agent Platform (AP). These are mandatory, normative components of the model. On the basis of FIPA specifications a line of platforms, in particular FIPA-OS [4], JADE [5], JAFMAS [1], JAF are created.

The majority of agent's platforms are created in Java [3] language. Java provides the advantages of architecture neutrality and portability for agent developers. The Java system solves the binary-distribution problem, implementing a "binary code format" that's independent of hardware architectures, operating system interfaces, and window systems. The format of this system-independent binary code is neutral to the architecture. If the Java run-time platform is made available for a given hardware and software environment, an application written in Java can then execute in that environment without the need to perform any special porting work for that application.

The advantage of Java platform is a uniform format of data representation on all hardware platforms. It means identical length of primitive data types, and also uniformity of their arithmetic rules.

The built - in multithreading support also is one of the basic advantages of Java platform that allows the agent to react simultaneously on a set of events of modeled system. Thus, parallelism of actions in uniform space of process is provided. The built - in classes of realization of streams contain all basic primitives and opportunities for efficient control of multithreading.

The remote method invocations (RMI) allow the programmer to create the distributed applications in which methods of the remote objects can be invocated by objects, which are placed on other virtual machines. There is an opportunity to generate programs, which are capable to move on a network.

2. Principles of design of self-optimizing MAS

Each agent in MAS should perform the certain set of actions supposing the use of resources and interaction with the other agents in its lifecycle. During interaction some difficulties at access to some resources because of their limitation can be observed. Therefore, there should be a mechanism of actions planning for the effective solution of a problem into system.

The mechanism of planning can be either centralized, or distributed. In the centralized algorithm of planning it is possible to take into account characteristics of elements in system more precisely, thus to realize more effective schedule. However, at failure of planning mechanism the system becomes not useful.

The distributed planning mechanism is more reliable at failures of separate system elements. Each agent independently plans the activity, proceeding from own targets and belief. However, at the distributed planning it is difficult to take into account effects of collective interaction in set of agents.

At present for development of agent's community the various algorithms based on heuristics are offered. These algorithms allow determining rules of agent's behavior, which promoting increased efficiency of their collaboration at the solution of a challenge. The heuristics have the various natures and are focused on use of knowledge of various agents' characteristics.

The algorithms of adaptive planning based on heuristics uses the modern methods of the optimization, among others genetic algorithms, neural networks and fuzzy logic. Such algorithms allow receiving the optimum solutions, which are implicitly taking into account features of cooperative activity of agents in concrete conditions. Adaptive planning and management in MAS giver opportunity to apply such systems successfully to the solution of various problems.

3. MAS optimization framework on the basis of virtual network algorithm

For optimization of the problem solution in MAS and the organization of adaptive planning in underlying MAS framework it is necessary to make some changes in framework components. First of all, it is required to determine characteristics that will reflect the increase of efficiency by MAS functioning. As such characteristics it is possible to choose the overall solution time, resources use parameters, directive terms (in real time modes), etc.

The second stage will be embedding in framework the tools of measurement of the chosen parameters. It can be carried out with implementation in the objects, representing the agents, the special profiling interface. This interface will allow to measure parameters of efficiency and to keep them in structure of profiled object.

At last, the measured parameters should be estimated in the optimization system. This system also is built in MAS framework and collects the information received at profiling during runtime. Then the optimization system estimates these data on the certain algorithm and offers recommendations on change of MAS configuration to improve its efficiency.

For effective introduction of profiling tools in MAS framework it is necessary to analyze a structure of the typical agent and its interaction with the basic components of framework architecture to designate places of embedding. The structure of the typical agent is shown on fig. 2 (based on JAFMAS model [1]).

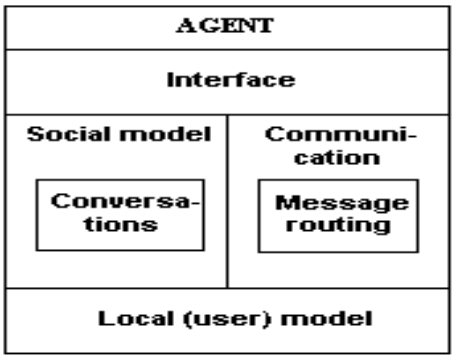

Fig. 2. A typical agent structure.

The agent is abstract object that can be expanded to the user object by definition of its local model. The local model describes the actions, which can be executed by object during search of the problem solution. Except the local model, the agent contains the social model determining control facilities for actions based on the conversations mechanism, and also the model of communications allowing agent to communicate with the other agents in system.

There will be the natural architecture in which the statistics about activity of agent will gather at a level of social model and communications. The abstract classes realizing this level, can be expanded with profiling interfaces, allowing to estimate duration and efficiency of execution of agent actions. Besides at the given level it is possible to enter mechanisms of more complex estimation of agent activity taking into account the several parameters.

After estimation of parameters about agent activity it is possible to apply optimization algorithm to whole system. As model for MAS optimization algorithm the model of a virtual associative network [6] is chosen. In this model set of cooperating elements is divided on set of clusters. Everyone cluster contains closely coupled elements between which the most effective interaction is established. During model training stage the optimal configuration of system having the maximal integral estimation is formed. The training a migration of elements of model in MAS structure, providing the best cooperation, can be used. The possible

specialized scheduling algorithms realized in units of the computing system are implicitly taken into account, because the integrated estimation takes into account the general parameters of quality the cooperation.

The algorithm of a virtual network can be realized by the centralized or distributed way. At the centralized realization it can be placed within the AMS system of framework. After each cycle of MAS functioning its reconfiguration is carried out. At the distributed way the agent determines for itself the best place in system, being based on own supervision.

Conclusion

Adaptive MAS optimization allows increasing essentially quality of search of solutions and brings a new level of intellectual behavior into systems. Using the modern technologies of optimization allows minimizing expenses for realization of such systems. The suggested approach will find his place at creation of MAS frameworks.

At present on the basis of the virtual network approach the experimental system of MAS modeling for studying characteristics and the further improvement of MAS adaptability is under construction.

References

[1]. D. Chauhan. JAFMAS: A Java-based Agent Framework for Multi-agent Systems Development and Implementation. PhD thesis, University of Cincinnati, 1997.

[2]. Developing multi-agent systems with a FIPA-compliant agent framework Fabio Bellifemine, Agostino Poggi, Giovanni Rimassa, in: "Software - Practice And Experience", 2001 no. 31, pp. 103-128.

[3]. http://java.sun.com

[4]. http://www.fipa.org/about/index.html

[5]. S.Poslad, P. Buckle and R.Hadingham, "The FIPA-OS agent platform: Open Source for Open Standards", published at PAAM2000, Manchester, UK, April 2000.

[6]. R.Kh. Sadykhov, A.V. Otwagin. Solution search algorithm of solution search for systems of parallel processing based on a virtual neural network model. // Avtomatika I vychislitel’naya tekhnika. - №1. – 2001. Riga, Latvia, pp. 25-33. (In Russian).