Proceedings of the *International Conference on Theory and Application of Mathematics and Informatics ICTAMI 2005* - Alba Iulia, Romania

A NOVEL INTELLIGENT MOBILE AGENT ARCHITECTURE

Barna Iantovics

Abstract. Many mobile agents don't have capabilities that can be associated with the intelligence. In this paper, we propose a novel intelligent mobile agent architecture. The proposed architecture combines the advantages of the mobile and static agents. A mobile agent with the proposed architecture may be endowed with capabilities of the intelligent static and mobile agents.

2000 *Mathematics Subject Classification*: Artificial Intelligence.

1. The mobile agents' intelligence

There are many definitions to the static agents' *intelligence* [5, 6, 13, 17]. The endowment of a static agent with intelligence in many cases increases his behavioural complexity and the size of the agent's body. The endowment of a mobile agent with the same intelligence as a static agent's intelligence increases the mobile agent body size and his behavioural complexity. The transmission of a large number of intelligent mobile agents in the network may increase the overloading degree of the network. A large number of intelligent mobile agents at a host may overload the host. These reasons demonstrate the difficulties to use intelligent mobile agents with the same intelligence as the static agents' intelligence. The mobile agents are usually assumed to have only a very limited or even no intelligence [1, 11, 14, 16, 2, 3, 4, 9, 18]. Many of the formal modeling of mobile agents is in terms of their mobility, they are not built upon a framework that explicitly supports the intelligent feature of the agents. Such models are typically reactive rather than pro-active. The mobile agents act in response to their environment, they are not able to exhibit goal directed behaviors. In many situations, the multiagent systems formed by cooperative mobile agents are considered to be intelligent [8, 12, 18, 19]. The mobile agents' intelligence is considered at the level of the multiagent system in which they operate. If the mobile agents cooperate they can solve intelligently difficult tasks.

2.The Proposed Mobile Agent Architecture

An *agent architecture* is essentially a map of the internals of an agent, its data structures, the operations that may be performed on these data structures, and the control flow between these data structures [17]. In this paper, we propose a novel intelligent mobile agent architecture. Some elements of the proposed architecture are described in the papers [6, 7]. A mobile agent *MOBILE AGENT* endowed with the proposed architecture is composed from two parts a static part *STATIC PART* and a mobile part *MOBILE PART*.

$$MOBILE\ AGENT = STATIC\ PART + MOBILE\ PART.$$

The static part of a mobile agent consists in a static subagent. The notion of *subagent* is defined in the paper [5]. The static subagent is responsible for the overtaking of the tasks from the user. The mobile part is composed from a variable number of mobile subagents. The static subagent creates the mobile subagents. The mobile subagents have all the proprieties of the known mobile agents. The mobile subagents are responsible for the tasks solving. The body of a mobile subagent contains the description of the agent's behaviour (the tasks solving methods) and may contain different data used in the tasks solving. Denote *specialization* the method that describes the solving of a class of tasks.

2.1.The Proposed Mobile Agents' Knowledge Bases

In the following, we analyze the knowledge bases of a mobile agent endowed with the proposed architecture. The knowledge bases of the mobile agent are detained by the static subagent. The static subagent has four knowledge bases denoted:*specializations*, *rules*, *network* and *clones*.

The *specializations knowledge base* contains a set S of specializations.

$$S = \{s_1, s_2, \ldots, s_n\}.$$

Each specialization s_i is a task solving method which allows the solving of the tasks, members of a class c_i of tasks. A mobile agent endowed with the proposed architecture can solve a set $C = \{c_1, c_2, \ldots, c_n\}$ of classes (types) of tasks.

The *rules knowledge base* is formed by a set of rules of the following form:

$$type \rightarrow specialization.$$

Where: *type* represent a task class, *specialization* represent the necessary specialization for solving the tasks members of the class *type*.

During the life cycle, the mobile agent can maintain the created mobile subagents that can solve efficiently some tasks. The *efficiency* (*performance*) of a mobile subagent can be measured by recording how fast the mobile subagent can solve the overtaken tasks. Performance evaluation of the mobile agents remains a difficult task, mainly due to the characteristics of mobile agents such as distributed and asynchronous execution, autonomy and mobility [10]. The same task solving at different hosts, using the same specialization, may be different. The hosts may have different tasks solving *capacities* and *capabilities*. A host's *capability* indicates the types of tasks that can be solved at the host by the mobile subagents. A host's *capacity* indicates the amounts of tasks that can be solved at the host. The *clones knowledge base* contains the maintained efficient mobile subagents obtained during the mobile agent operation.

In the clones knowledge base the knowledge about each mobile subagent has the following form:

$$ms =< I, H >, I = (h_{i1}, h_{i2}, \ldots, h_{ik}), H = (H_{i1}, H_{i2}, \ldots, H_{ik}).$$

Where: ms represents the identifier of the mobile subagent, I represents the itinerary of the mobile subagent, h_{i1}, h_{i2}, $\ldots$, h_{ik} represent the hosts that must be visited by the mobile subagent, H describes the specializations used at each host, $H_{ij} = < h_{ij}; s_{r1}, s_{r2}, ..s_{ru} >$ specify the specializations s_{r1}, s_{r2}, $\ldots$, s_{ru} which can be used at the host h_{ij} at the tasks solving.

The *network knowledge base* contains different knowledge about the network where the mobile agent operates. As examples of knowledge, which can be detained in the network knowledge base we mention: the information about some hosts from the network, the topology of a part of the network, latency, bandwidth etc. In the case of a host, the network knowledge base may contain information like: the tasks that can be solved at the host, the intervals of time when the host is usually overloaded etc.

2.2.The Proposed Mobile Agents' Operation

A tasks solving cycle consists in overtaking and solving the tasks transmitted by the user. In the following, we describe briefly the *algorithm intelligent mobile agent* that illustrates the manner in which a mobile agent endowed with the proposed architecture operates at a tasks solving cycle.

Algorithm Intelligent Mobile Agent

Step 1 The tasks overtaking from the user.

The static subagent overtakes the tasks T transmitted by the user.

$$T = \{t_1, t_2, \ldots, t_k\}.$$

Step 2 The establishment of the tasks solving.

The static subagent establishes the useful mobile subagents CMS from the clones knowledge base which can solve a subset TS of the overtaken tasks T.

$$CMS = \{cms_1, cms_2, \ldots, cms_g\}.$$

The static subagent establishes the necessary mobile subagents MS which must solve the set T-TS of tasks.

$$MS = \{ms_1, ms_2, \ldots, ms_j\}.$$

While ($CMS \neq \emptyset$) *do*
{

The static subagent executes the following actions:

- picks up the next unlaunched mobile subagent cms from the set CMS;
- endows the mobile subagent cms with some tasks from the overtaken tasks;
- locates the first host specified in the itinerary of the mobile subagent cms. It transmits the subagent to the located host.

$CMS := CMS$ - $\{cms\}$.
}

While ($MS \neq \emptyset$) *do*
{

The static subagent selects an uncreated mobile subagent ms from the set MS.

The static subagent creates and launches the subagent ms which consist in:

- establishing the itinerary I that must be traversed by the mobile subagent ms. It places the itinerary to the mobile subagent ms;

- establishing the tasks from the set T-TS that must be solved at each host specified in the itinerary I. In the case of each overtaken task establishes the necessary specialization. Endows the mobile subagent ms with the established tasks and specializations;

- creating a clone cls of the mobile subagent ms;

- locating the first host specified in the itinerary I. Transmits the mobile subagent ms to the located host.

$MS := MS$ - $\{ms\}$.
}

Step 3 The overtaken tasks solving.

Each launched mobile subagent executes the following actions at each visited host:

- solves the tasks, the results obtained are transmitted to the static subagent;

- the specializations that are not necessary in the following tasks solving are eliminated.

Step 4 The solutions transmission to the user.

The static subagent collects the results transmitted by the mobile subagents. If it is necessary the static subagent can process the results, the obtained solutions are transmitted to the user.

Step 5 The surviving mechanism.

The clone agents of the efficient mobile subagents are retained in the clones knowledge base.

End.

At the beginning of a tasks solving cycle, the static subagent overtakes the tasks T transmitted by the user. The user in the following form transmits each task:

$$< type, description, priority, deadline > .$$

Where: *type* represents the type of the task, *description* represents the description of the task, *priority* represents the priority of the task, *deadline* represents the maximum time allowed to the task solving.

The static subagent can picks up autonomously the necessary specialization to a task solving. Each task type has associated a task solving specialization. The selection of a task solving specialization is polynomial in complexity. Each transmitted task has associated a task type. When the static subagent receives a task, he checks in the rules knowledge base the rule whose precondition is the task type, the selected rule postcondition defines the specialization that must be used for the task solving.

After the tasks overtaking, the static subagent creates mobile subagents, some used mobile subagents can be from the clones knowledge base. A mobile subagent from the clones knowledge base is selected; if corresponding to each specialization of the agent can be associated a task from the overtaken tasks that can be solved using the specialization. A mobile subagent selection from the clones knowledge base consists in creating a copy of the agent that will be used in tasks solving, the original agent remains in the clones knowledge base. The endowment of an agent from the clones knowledge base consists in completing the agent body with the tasks that must be solved, to each task the task solving specialization is associated. The mobile subagents CMS selected from the clones knowledge base will solve a subset TS of the overtaken tasks T. The tasks T-TS that are not overtaken by the mobile subagents CMS must be solved by newly created mobile subagents.

The mobile subagents operate as the known mobile agents. Each launched mobile subagent is responsible for the solving of one or more tasks. Each overtaken task by a mobile subagent is solved at a host, the host executes the tasks solving description from the mobile subagent body. The mobile subagents transmit the tasks solutions after they are obtained. Before the launch of a newly created mobile subagent, the static subagent creates a clone of the mobile subagent, which represents a copy of the mobile subagent. A clone mobile subagent doesn't contain the overtaken task description by the mobile subagent, only contains the itinerary and the used specializations by the mobile subagent whose copy it represents. After a tasks solving cycle the static subagent maintain the created efficient clone mobile subagents. The clone mobile subagents with lower efficiency are eliminated.

A mobile subagent eliminates during his work the specializations that are not necessary in the next tasks solving. In this way, the data transmission in the network is lower. It is not necessary for a mobile subagent to return to the static subagent. If a mobile subagent is efficient, the static subagent can use the clone of the mobile subagent in the next tasks solving cycles. At the endowment of a mobile subagent with specializations some specializations may be absent. The absent specializations are transmitted to the mobile subagent just when and where they are necessary. The subagents of the mobile agent can communicate [7]. If a mobile subagent needs a specialization at a host, and the specialization is not present in his body, then the mobile subagent must query the specialization from the static subagent. This manner of operation is useful when the mobile subagent must visit numerous hosts, the specializations size is large and the mobile subagent needs the specialization at few hosts.

3.Simulation Results

We have realized simulations of a mobile agent endowed with the proposed mobile agent architecture using a developed simulation environment. The simulations' purposes were to compare the performance of the traditionally used mobile agents with the performance of the mobile agents endowed with the proposed mobile agent architecture. A proposed mobile agent may uses more mobile subagents at each tasks solving cycle in the overtaken tasks solving. The performance evaluation consists in the necessary time for solving all the overtaken tasks at a tasks solving cycle. The simulation environment parameters consist in the simulated hosts and the specializations of each host. Each host is endowed with a specializations set.

The simulation environment allows the simulation of the tasks solving by the mobile agent endowed with the proposed architecture. For each generated mobile subagent the nearly optimal itinerary in the overtaken tasks solving time using an evolutionary algorithm is determined. A mobile subagent itinerary consists in the hosts that must be visited by the subagent and the tasks that must be solved at each host. Using the evolutionary algorithm different itineraries are generated in the case of each created mobile subagent. The best mobile subagents' itineraries obtained during the searching process represent the solution (the itinerary of each used mobile subagent is determined). A mobile subagent itinerary evaluation consists in the simulation of the overtaken tasks solving by the mobile subagent that has the itinerary. The evaluated performance of a mobile subagent itinerary consists in the necessary time for solving all the overtaken tasks by the subagent.

Figure 1 and Figure 2 present simulation results of a mobile agent endowed with the proposed architecture. In the simulation environment are used a set of 100 specializations distributed between 30 hosts. The simulations were realized for different sets of overtaken tasks with different necessary solving time and different overloading degree of the hosts.

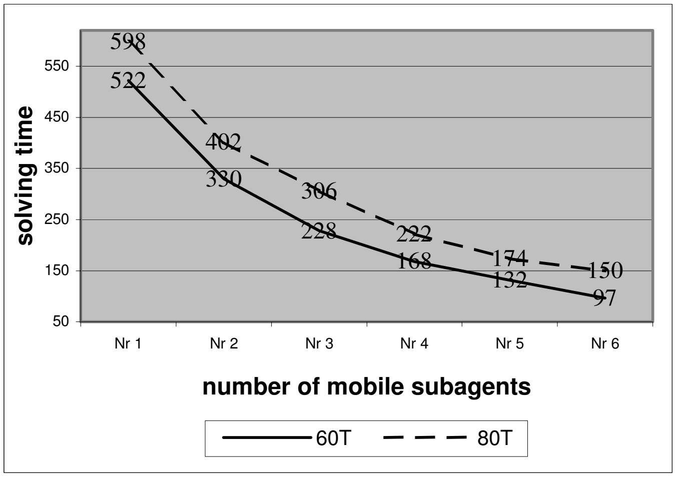

Fig.1. Simulation results of a proposed mobile agent

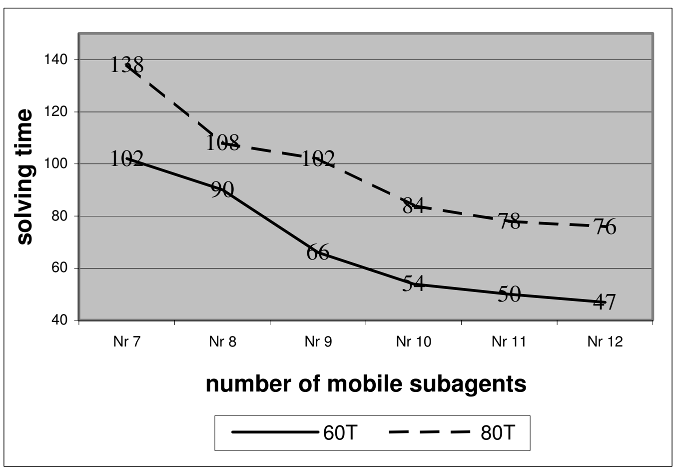

Fig.2. Simulation results of a proposed mobile agent

Figure 1 illustrates how the average tasks solving time measured in seconds of 60 and 80 overtaken tasks is changing using from 1 to 6 mobile subagents. Figure 2 illustrates how the average tasks solving time measured in seconds of

60 and 80 overtaken tasks is changing using from 7 to 12 mobile subagents. The results obtained during the simulations illustrate that more mobile subagents can solve faster a large number of overtaken tasks than a single mobile subagent. How many mobile subagents are necessary in solving efficiently a set of overtaken tasks depends on the network and hosts resources overloading degree and the parameters of the tasks (number, difficulty) which must be solved.

Figure 3 illustrates how many times the tasks solving time is improved in average using a mobile agent endowed with the proposed architecture that uses from 2 to 7 mobile subagents in the solving of 50 and 80 tasks as opposed to a traditionally used mobile agent used in the same conditions. The simulations were realized for different number of hosts endowed with different specialization sets, different overloading degree of the hosts and different sets of overtaken tasks. The simulations show that a mobile agent endowed with the proposed architecture that uses more subagents can solve faster a large number of tasks that must be solved at more hosts than a traditionally used mobile agent.

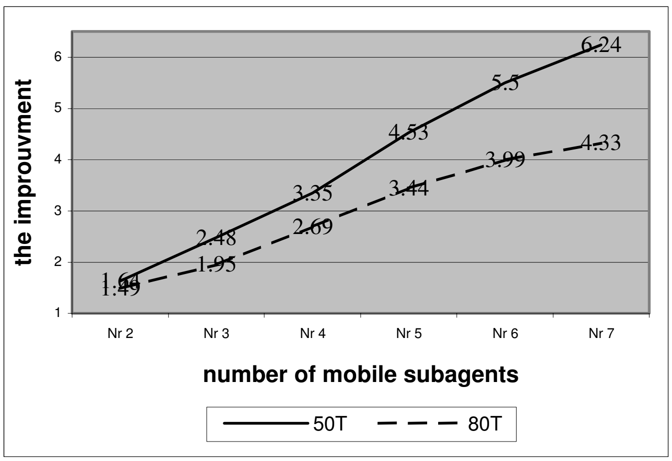

Fig.3. A mobile agent versus a proposed mobile agent

4.The proposed mobile agents' intelligence

In the following, we present motivations which illustrate the intelligence of the proposed mobile agents. A mobile agent with the proposed architecture can be endowed with capabilities of the intelligent static and mobile agents. The static subagents of the mobile agent may have any sort of intelligence characteristic to the static agents like [5, 13, 17]: flexibility in the operation and learning capability.

The mobile agent is flexible in operation. The static subagent may have a large number of specializations, which allows the solving of a large variety of heterogeneous tasks. It can endow a mobile subagent with any of his specializations. The static subagent may overtake a large number of tasks from the user. Each task is solved when it is possible. The tasks that cannot be solved at a tasks solving cycle are postponed until the next tasks solving cycle, when the postponed tasks solving processes are reloaded. If a task is postponed, then its priority and deadline is increased, which guarantees that the task is solved and the task solving is not postponed for a long time.

The subagents of the mobile agent can communicate [7]. The static subagent can be endowed with learning capability. Mobile subagents may transmit experiences to the static subagent. The static subagent can learn from these experiences. The static subagent may use any learning technique that can be used by a static agent. In this way, the mobile agent can learn new specializations or it can improve the efficiency of the existent specializations.

5.Conclusions

The mobile agents can be considered a relatively new paradigm in the area of distributed programming and a useful supplement of traditional techniques like the Client/Server architecture. Mobile agent technology has been applied to develop the solutions for various kinds of parallel and distributed computing *problems* (*tasks*). In this paper, we have proposed a novel mobile agent architecture. The novelty of the proposed agent architecture consists in the combination of the mobile and static agent paradigm. The proposed mobile agent operation is different from the agent cloning presented in the paper [15]. A static subagent creates during his life cycle mobile subagents with different specializations sets. The created mobile subagents are endowed efficiently with specializations. A surviving mechanism is also adopted, from the created mobile subagents are retained in a knowledge base, the agents whose efficiency is practically demonstrated, this agents can be used in the future.

The main advantages of the proposed mobile agent architecture are the: mobility and intelligence. A mobile agent with the proposed architecture can be endowed with many capabilities that can be considered a component of intelligent behaviour. The paper [7] analysis the possibility to create intelligent cooperative multiagent systems with agents endowed with the proposed agent architecture.

References

[1] G. Cabri, L. Leonardi and F. Zambonelli, *Engineering Mobile-Agent Applications via Context dependent Coordination*, Proceedings of the 23rd International Conference on Software Engineering (ICSE 2001), Toronto Canada, 2001, 371-380.

[2] C. Erfurth, P. Braun and W. R. Rossak, *Migration Intelligence for Mobile Agents*, Artificial Intelligence and the Simulation of Behaviour Symposium on Software mobility and adaptive behaviour (AISB 2001), University of York, United Kingdom, March 2001, 81-88.

[3] C. Erfurth, W. R. Rossak, *Autonomous Itinerary Planning for Mobile Agents*, Proceedings of the Third Symposium on Adaptive Agents and Multi-Agent Systems (AISB 2003), University of Wales, Aberystwyth Great Britain, April 2003, 120-125.

[4] C. Erfurth, W. R. Rossak, *Characterization and management of dynamical behaviour in a system with mobile agents*, Proceedings of the Innovative Internet Computing System - Second International Workshop (IICS 2002), Khlungsborn Germany, LNCS 2346, (2002), 109-119.

[5] J. Ferber, *Multi-Agent Systems, An Introduction to Distributed Artificial Intelligence*, Addison Wesley, 1999.

[6] B. Iantovics, *Intelligent Mobile Agents*, Proceedings of the Symposium "Zilele Academice Clujene", Babes-Bolyai University of Cluj-Napoca, 2004, 67-74.

[7] B. Iantovics, *A New Intelligent Mobile Multiagent System*, IEEE International Workshop on Soft Computing Applications (IEEE - SOFA 2005), Szeged-Hungary and Arad-Romania, August 2005 (accepted paper).

[8] R. Kowalczyk, P. Braun, I. Mueller, W. Rossak, B. Franczyk and A. Speck, *Deploying mobile and intelligent agents in interconnected e-marketplaces*, Journal of Integrated Design and Process Science, Transactions of the SDPS, vol. 7, no. 3, (2003), 109-123.

[9] H. Ku, G. W. Luderer and B. Subbiah, *An Intelligent Mobile Agent Framework for Distributed Network Management*, Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM'97), Phoenix USA, November 1997.

[10] X. Li, J. Cao and Y. He, *A Direct Execution Approach to Simulating Mobile Agent Algorithms*, The Journal of Supercomputing, Kluwer Academic Publishers, The Netherlands, vol. 29, (2004), 171-184.

[11] C. Mascolo, *MobiS: A Specification Language for Mobile Systems*, Proceedings of the Third International Conference on Coordination Models and Languages, Amsterdam, The Netherlands, April 1999, In: Ciancarini, P. and Wolf, A. (eds.), Springer-Verlag, LNCS 1594, (1999), 37-52.

[12] S. McGrath, D. Chacn and K. Whitebread, *Intelligent Mobile Agents in the Military Domain*, Fourth International Conference on Autonomous Agents, June 2000.

[13] R. Pfeifer and C. Scheier, *Understanding Intelligence*, MIT Press, 1999.

[14] G. C. Roman, P. J. McCann and J. Y. Plun, *Mobile UNITY: Reasoning and Specification in Mobile Computing*, ACM Transactions on Software Engineering and Methodology, vol. 6, no. 3, (1997), 250-282.

[15] O. Shehory, K. P. Sycara, P. Chalasani and S. Jha, *Agent cloning: an approach to agent mobility and resource allocation*, IEEE Communications Magazine, vol. 36, no. 8, (1998), 58-67.

16] A. R. Silva, A. Romo, D. Deugo and M. M. da Silva, *Towards a Reference Model for Surveying Mobile Agent Systems*, Autonomous Agents and Multi-Agent Systems, vol. 4, no. 3, (2001), 187-231.

[17] G. Weiss, (ed.), *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, MIT Press, Cambridge Massachusetts London, 2000.

[18] H. Xu, *A Model-based approach for development of multi-agent software systems*, Ph.D thesis, the Graduate College of the University of Illinois at Chicago, 2003.

[19] K. Yang and A. Galis, *Rule-driven Mobile Intelligent Agents for Real-time Configuration of IP Networks*, Proceedings of Knowledge Based Intelligent Information and Engineering Systems, (KES 2003) University of Oxford 2003.

Barna Iantovics
Department of Computer Science
Petru Maior University of Tg. Mures
Str. Nicolae Iorga, Nr.1 Tg.-Mures, Romania, 540088
email:*ibarna@davos.science.upm.ro*