USE OF MULTI-AGENT SYSTEMS IN PROJECT MANAGEMENT

Martin Dosedla
Doctoral Degree Programme (2), FIT BUT
E-mail: xdosed03@stud.fit.vutbr.cz

Supervised by: Jitka Kreslíková
E-mail: kreslika@fit.vutbr.cz

ABSTRACT

This document describes the use of MAS (multi-agent systems) in the field of project management. In the application of adaptive management processes can be applied technology and the possibility of MAS from the beginning of the life cycle. A lot of areas of software project management can be deal with using agents. Scheduling, development plans, scenarios solution and especially risk management. This article summarizes the possible usage of such approach.

1. INTRODUCTION

Multi Agent Systems are designed for better understanding and control of many different types of projects and tasks. Advantages of MAS approach [1]:

- increasing flexibility, adaptability and autonomy
- load leveling to solve complex and large-scale problems

The part of software development – project management (PM) – is typical application area for this approach. In domain of PM is necessary to apply the newer knowledge and possible ways of modern approach. With MAS comes reducing centralism and workload sharing.

Another reason for use adaptive agents in areas of planning and management is a different approach. The whole complex plan is not known, but views (sub-plans) are gradually creating and adapting to the current situations. This situation is monitored by the agents on the basis of their knowledge and perception of the environment (including other agents in system) [3]. The whole MAS can create a strategy for management.

In the following paragraphs we will discuss the main aspect of the planning and scheduling with respect to risk management.

2. PLANNING AND SCHEDULING IN PROJECT MANAGEMENT WITH MAS

The use of distributed artificial intelligence, on which MAS is based, can in the first place bring us a different look at planning for the application of project management. In a classic application of knowledge based planning, we build a comprehensive plan to achieve the objective. This is a similar system, as well as rational agents in the classic MAS. However, we can use a slightly different principle.

On the basis of the principle of the BDI (beliefs - desires - intention), we can separate and balance selection of activities from the doing of a plan [3]. Decision making depends on the agent's private knowledge and ideas but also on communication with neighboring agents. Described architecture of PRS (Procedural Reasoning System) is also based on the principles of the BDI. For our purpose, there is an important difference between the classical approach of planning and the PRS with MAS. In classical planning occurs generating of a complete plan to achieve the objective at the beginning of the cycle (this is the traditional principle of planning in AI). If there are any changes in the course of implementation, whole plan must be re-completing. It’s more problematic and less flexible to the response and reaction to changes. The flexibility and adaptability are very advantageous properties in the management of software processes (and elsewhere). In the PRS systems is balanced planning with the executing of those plans. In one moment it does not know the whole plan, but only partial specifications of the plan for the future. In accordance to the principle of the BDI - agents in the PRS act to achieve the objectives.

The Interpret of such architecture coordinates beliefs, desires and intentions. And it also works with databases of partial plans for the transformation from initial state to the final state. Database of plans is the field of knowledge of agents. Plans are selected with regard to the intention of the agent and the state of the environment.

Therefore planning is in progress operationally - during the actual running of the software process, which brings dynamism and adaptability. Some basic knowledge of the planning process (e.g. completion of the necessary data, etc.) must be surely set at the start of software project. These skills are in place prior to the knowledge base of agents (beliefs). All the planning and executing actions are implemented with regard to them.

Decision-making and collecting plans are included in the BDI principle. For the creation of plans, it would appear appropriate decomposition system activities to sub-activity. These sub-activities are the sequence of steps to achieve this objective. Each agent can have a target (partial goal) and will act to achieve it. For the MAS exists a key plan of the group. Its creation is based on negotiation between agents. The goal is to create a plan that is right (at a time, at the price of and with the available resources).

3. RISK MANAGEMENT WITH MAS SUPPORT

One of the key areas in the management of software projects is risk management. The system of manage activities, reduce and address risks and their impact on the project. These activities are repeated and linked in the implementation of the project. As a result, there are the four phases: identification, evaluation, control strategies and monitoring [2].

With using the MAS, the risks must be (as soon as possible) identified and solving. MAS is a system with multiple agents, thus it’s important a communication between them and social ties. Apart from the definition of common language (ACL - Agent Communication Language) and perception, the major-role play task of determining the objectives of individual agents. In general MAS for project management, software agents may have different targets. These targets are only partial coherent. The communication will include cooperation or competition among the intentions of agents in a system. Agents have their own dominant strategy, which is independent on other agents. Finally the outcome common strategy of whole group of agents must be chosen. In the fact the agents created a group (a coalition agreement) and will present commitments to cooperate.

In the case of identification a risk by the agent, the commitment between agents to identify risks and provide information for its solution may be agreed. Then agents must negotiate the commitment to inform about achieves the objective of finding the correct action against the identified risks. The risks may occur at any time during the management process. The first step to successful risk management is a good knowledge base and intentions [2]. This occurs in the implementation phase of project management system based on agent approach. Agents solving risk situations in line with the BDI.

We can sum up those four phases of risk management and their application to the PRS. The identification of risks will take place as soon as possible, when a knowledge base of risks is created. Risks can be identified during the whole project life cycle. That risk rating is based on the knowledge-base, with specified probabilities of the impact on the project. Rating of risks is also affected by the reasoning of agents. Managing of risks is carried out in commitments to other agents. For define strategy of managing risks, workflow management and communications between agents is set. Monitoring of risks realized directly on the basis of assessment of the state agent sensors in accordance to the intentions.

Risk management is one of the important areas in project management. Complete life cycle of the software development process can be assisted and described by use software agents (Figure 1).

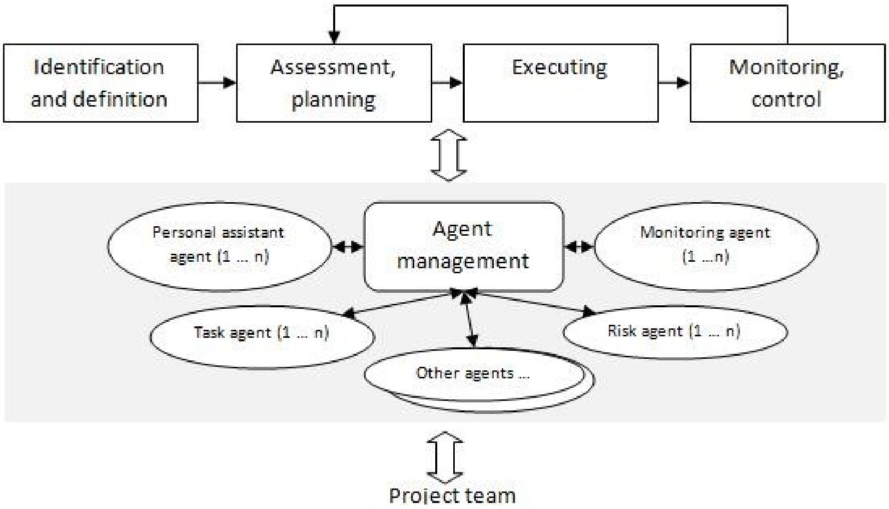

Figure 1: MAS for software project management

Agents may, therefore, bring benefit practices not only in the phase of project planning, but also in the main control phase of sub-project activities. Risk management is a critical point in the field of software projects.

4. PRACTICAL APPLICATION OF THE PRINCIPLES

Program agents (soft-bots) may be implemented by more ways. For our purpose (software project management) will be undoubtedly specialized agents for this specific role. Rational agents have planning and executing units for its activities. These units act on the base of knowledge and reflection [3]. Architecture of rational agent is formed (correspondence with the PRS and BDI principles) by the following units (Figure 2):

- Unit for communication with the environment formed by Sensors and Actuators. Communication is between agents and environment status.

- Resource management and knowledge area - library of plans and scenarios and their control. Knowledge area is formed at the start of project and adapted during process.

- Planner and Scheduler – units based on the principle of the BDI for planning and scheduling of project tasks.

- Workflow interpreter – the main control unit coordinating activities of the agent.

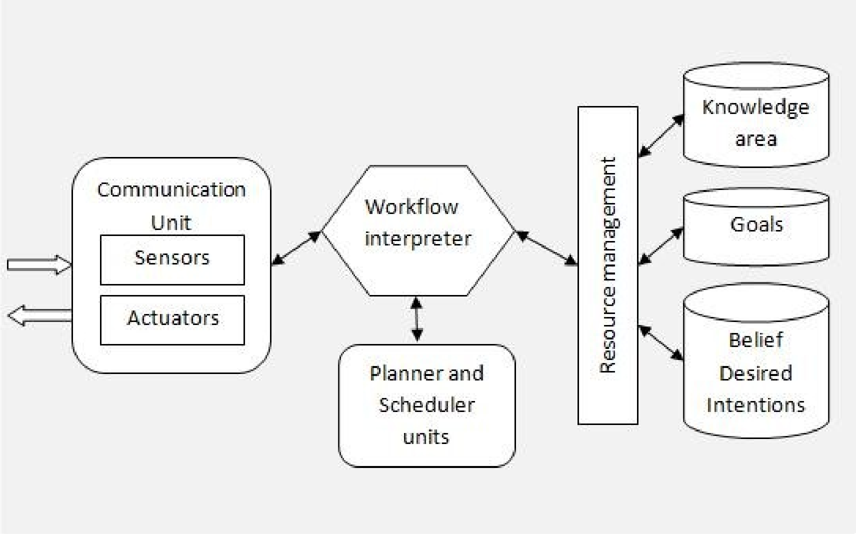

Figure 2: Architecture of agent in management system using MAS

PRS agent works in a loop. Agents receive sensations from the environment. On their basis agents adapt the internal model and select the intentions. From those knowledge agents draw up a plan for the next exercise.

BDI and PRS theoretical principles can be used in a number of implementation tools. Most tools are implemented on Java language base. Tools like JADE, JAM, Retsina, UMPRS and others are some of them. Most instruments support top-down, goal-based or bottom-up data driven reasoning [1]. Collection and selection goals and plans are usually based on the maximum priority. It's important the support for the simultaneous executing of more goals in the system. Agents can have different priorities of goals. Also planning and construction of plans is one of the most important properties. For the creation of quality systems are implemented sequencing, iteration, sub-goaling, atomic (i.e., non-interruptible) blocks, n-branch deterministic conditional execution, explicit failure-handling section, and others in systems.

Quality of implementation depends on the selection of a suitable environment and system. Although the majority of systems are based on the principles of the BDI, the implementation and potential of development tools are not always the same. Knowledge, goals and plans, as essential part of the system, are used to achieve the overall target state [1].

For the application of software project management are important these aspects and possibilities in environment:

- Determination of detailed plans and sub-plans and conditions for their exercise.

- The actions for plan-failure. Negotiations with other agents.

- The definition of priorities and detailed plans of action to control their implementation.

- On the basis of the exercise plans and monitoring status, risks must be controlled and solved in time (as soon as possible).

- It's necessary to enable to correct the plans on accordance to actual progress of a project. Changes are controlled by testing of environment of MAS.

Aforementioned environments and development tools meet the following conditions (although their implementation is quite different). It's possible to use them in this area for the control of processes in the software products development.

5. CONCLUSION AND FUTURE WORKS

In this paper we described some of the principles and application of agents and multi-agent systems. We showed the main advantages in the area of software project management. Use of this approach should lead to more effective management and increasing the adaptability of the process. Adaptability to the risk situations is important property of management system. Currently the issue is further investigated by us. We will select techniques and environment to implement some characteristics of MAS management system to verify the possibility of MAS in this field.

ACKNOWLEDGEMENTS

This research was supported by the Research Plan No. MSM, 0021630528

Security-Oriented Research in Information Technology.

REFERENCES

[1] Netrvalová, A. Úvod do problematiky MAS. ZČU v Plzni, FAV 2005 [online] URL: <http://www.kiv.zcu.cz/~netrvalo/>.

[2] Ulieru, M. Adaptive Risk Management System (ARMS). In: Network and Information Security Workshop, Halifax, 2006.

[3] Huber, M. J. Intelligent Reasoning Systems. [online] URL: <http://www.marcush.net/IRS/>

[4] Badr, R., Reilly, D., TelebBendiab, A. A Conflict Resolution Control Architecture For Self-Adaptive Software. School of Computing and Mathematical Science, Liverpool John Moores University 2002.

[5] Nienaber, R. C., Barnard, A. Software Quality Management Supported by Software Agent Technology. University of South Africa, Pretoria, South Africa 2005.

[6] Laddaga, R., Robertson, P. Self Adaptive Software: A Position Paper. 2004.

[7] Rajyaguru, Ch., Motwani, N. An Architecture-Based Approach to Self-Adaptive Software. University of Southern California. 2003.

[8] Berry, P. K., Myers, K. L. Adaptive Process Management: An AI Perspective. Artificial Intelligence Center, California 1998.