ADAPTIVE PROCESS MANAGEMENT OVERVIEW

Martin Dosedla
Doctoral Degree Programme (1), FIT BUT
E-mail: xdosed03@stud.fit.vutbr.cz

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

ABSTRACT

This document provides an overview of adaptive process management. Complex business processes require dynamical management of a development. The environment for creation software products is complex, dynamic and frequently distributed. These characteristics need advanced process and workflow management. Adaptive management that uses software agents offers this. Intelligent multi-agent systems adapted to management area deals better with entire process. Multi-agent systems (MAS) can be applied is several areas and several ways to improve process of creation software. This paper summarizes how useful can adaptive process management be.

1. INTRODUCTION

Development of many kinds of product needs a lot of sources. Software product requires time, money and other resources to reach the goal. Some years ago software product was isolated. Nowadays there is need for more complex software product. More interconnected modules cooperate together. More users use the program. And everyone have different needs. Thus applications now are more sophisticated and often interconnected each other (especially in web or network environment). We can talk about adaptive (or self-adaptive) software. This software, the development and business processes are complex and not easy for management. To improve process and workflow management we can use adaptive approach. Agents and multi-agent systems - MAS - can help us several ways. Workflow management system (WFMS) is process-oriented task. In next paragraphs we can concentrate on the adaptive software and mainly on the adaptive process management.

2. ADAPTIVE SOFTWARE

Develop of software is expensive and takes more time. There was designed several ways how to reach the goal [7]. Three approaches are this:

- Classical structured programming – It's suitable for good designed applications. We need to know the plan and the requirements well. And before we start programming we have all use cases and complete specification that rarely changes.

- Object-oriented paradigm brings some positives to build complex software systems. We can now reorganize and reuse some object. And objects are encapsulated with limited interactions with others. From this point of view changes are simpler. But still we need, if specifications change-over, complete life-time cycle to rebuild the application.

- The third approach is Adaptive programming. If we use this scheme our software can adapt to new conditions. Some changes in operational environment or users' needs dynamically changes software behavior and actions [6]. The software is self-adaptive if can improve behavior as a respond to change inputs, internal or external conditions. Object is the main structural unit in object-oriented software. In adaptive or self-adaptive software we can use agents as a higher abstract unit. We can use deliberative agent architecture. Agents preserve representation of the environment and internal states. Then agents construct plans with priority criteria. Internal structure of the agent can be programmed potentially also with object-oriented languages. In agent and mainly MAS construction, we must use some training and machine learning techniques. Genetic algorithm and evolution techniques help us to construct adaptive software. MAS has base of knowledge and autonomous behavior and can act with minimal user interference.

The way how to create adaptive software is agent approach. Software agent is an entity or compute process [9]. The main attributes is autonomous behavior. So agent exists in complex environment and with sensors and effectors put into effect actions to reach the goal. High potential is more agents in the environment. We can talk about multi-agent systems. The environment can be possible decentralized and distributed with all its benefits. There exist several kinds of agents with special tasks.

In MAS agents cooperate and coordinate actions. This communications must be strictly defined. With Agent communication language (ACL) messages can be in several classic languages like XML, C++, Java or Prolog. Implementation of adaptive software can be difficult. There are some frameworks that can help us (for example JADE – Java Agent Development framework). Ideals for this target are dynamic programming language (DPL). With DPL the program can be change when running (for example can be added some method). But still there are risks in management of adaptive software (AS). Many properties are important – safety, adaptability, dependability, consistency, robustness, availability, correctness and others. For building AS we need not only agent technology. Methods of artificial intelligent, social networks, decision or probabilistic theory are also useful.

Self-adaptive software is useful for dealing with many kinds and forms of software. Very useful is in control, sensors, robotics, etc. But we can use an adaptive architecture with MAS also for operating and information systems and for management, planning, scheduling, configuration, risk or security systems. These adaptive systems can help us for develop and manage business processes.

3. ADAPTIVE PROCESS MANAGEMENT

Development in Software Project Management continuously changes and improves. Many approaches are commonly used in practice. Software is changing to distributed collaborative projects based on network technologies.

Adaptive process management describes the life cycle of software systems. This approach is effective for adaptive software as well as other types of software. Software Project consist a lot of processes. We can describe all steps of process that leads to complete or partial fulfillment as a workflow. Adaptive workflow management (AWFM) has these characteristics:

- Support dynamic distributed complex processes.

- Monitoring processes and environment and adapt processes.

- Plan and schedule agent and respond to agent activities.

The agent approach is needed in three areas:

- Definition of workflow process – based on knowledge about specific software project.

- Control and repair of workflow process – as a react to changes in environment or in current workflow processes.

- Reactive scheduling activities of agents – reactive agents (or behavior agents) take plans and responds to environment and other agents in MAS.

The adaptive and reactive control has been motivated primarily in control systems of physical mechanism and devices (automation, robots, …) [8]. But we can benefits from this in control processes. The software agent development covers science segments like AI, social networks, psychology, etc. And how we can use MAS in project management and AWFM? There are three different approaches [4]. MAS as a facilitate subsystem, MAS covering complete management, or MAS as a modeling and enacting technology. Of course combination of these approaches is possible but complicated. Now we can describe briefly all three of these approaches.

In the first we use agents only as an add-on subsystem for some existing process or workflow management system or approach. Different kinds of agents acting between existing management system and user interface. Agents help to user with some part of management (quality management, risk management, etc.). This approach allows use agents and some adaptive skills with existing management software. But many of benefits of MAS are unused.

The second approach covers MAS as a domain of process management. Different specific types of agent are uses to different tasks of process management in different roles. Specially designed agents help to solve problem of risk management, security, resource management, etc. In agents (and therefore in management system) are implemented strategies for specific management domain. Agents have a task and the system is more role-based than process-based. We thing this approach is possible to use in management but dynamic and adaptive process management need more.

Finally there is the third approach. We can use agent technology in the first place as a modeling technique. Agent based modeling can represent all entities in process oriented approach. When we have all processes modeled in agent based framework, we can use this information to construct agents. Derived agents give us support for enacting of processes. Risk, resource and others managements supported by specialized agents can be effectively used in distributed and dynamic processes.

Thus we see all three approaches have their plus and minus. But the third approach is for us systematic. We can use MAS in both directions:

- Modeling software and management processes, readjustment, customize and adapt processes.

- Enacting in processes, mechanisms for reactivity (ability to adapt behavior based on changes in environment) and dynamical control.

In Agent-based Adaptive Process Management System (ABAPMS) figured several types of agents. Different kinds of managements need specialized kinds of agents. The main properties are the same for all agents:

- Autonomy without human intervention others agents interactions and "intelligent" behavior.

- Social ability to communicate, coordinate and cooperate with other agents (emergence). Interact with humans.

- Dynamical reactivity and adaptability in changing environment.

Furthermore agents must be able act to reach the main goal. Deliberative agents in BDI theory (belief – desire – intention) act in cycle and adapt the plan to all new situations in environment.

For example in resource management we can use three mainly types of agents. Process agent is for workflow management. Repository agent control documents and source code repository. And resource agent to manage all types of resources needed in process [4]. Of course there must be also others types of agents – like task, tool, personal agents, etc.
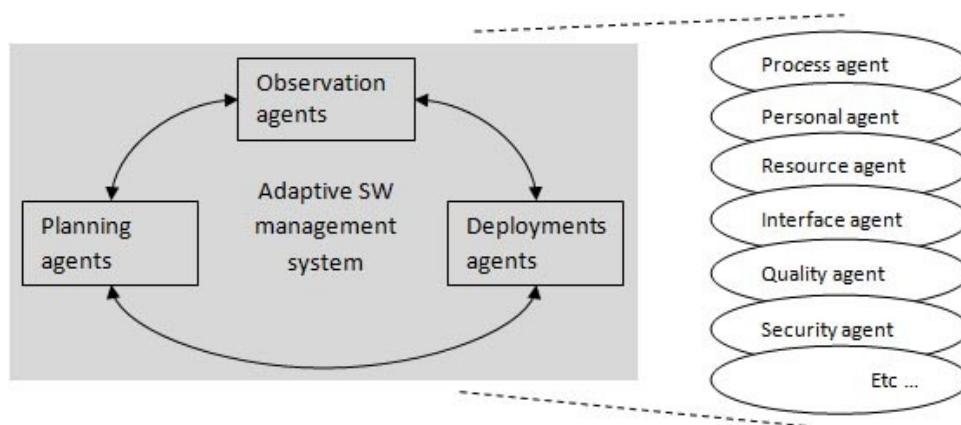
Figure 1: Adaptive software management system.

Generally we must use agents in these three ways (Figure 1). The main-core of ABAPMS is constructed from agents to observe, plan and deploy [1]. All agents cooperate (we can use ACL for transfer defined messages between agents) to the goal.

In management area of software project are risks. As we suggested in distributed environment that dynamical changes we (or our software) must react. With classical process or role oriented management changes are problem that cost time and money (or potentially prestige of company). If Agent-based Adaptive Process Management System is used, we can adapt to new conditions more quickly.

4. CONCLUSION

In this paper we have described potential and characteristics of the adaptive project and process management. For this task we can use software agents. This can really help us to control workflow in processes. In the modern distributed environment we must adapt management and development processes to the new situation. And this described system can help us with this no easy task.

ACKNOWLEDGEMENTS

REFERENCES

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

[2]   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.

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

[4]   Joeris, G., Klauck, Ch., Herzog, O. Dynamical and Distributed Process Management based on Agent Technology. In 6th Scandinavien Conference on Artifical Intelligence. 1997.

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

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

[7]   Norvig, P., Cohn, D. Adaptive software. Harlequin Incorporated, California. PC AI Magazine 1997.

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

[9]   Kubík, A. Agentově-orientované inženýrství: nové paradigma pro tvorbu softwaru? UI Slezská univerzita. 2003.