MODELING AND EXECUTING SOFTWARE PROCESSES
BASED ON INTELLIGENT AGENTS

Mlle F. Aoussat
Computer Science Department/ Faculty of Sciences
SAAD DAHLAB University, BLIDA
BP 270, Blida, Algeria.

M’M. Ahmed Nacer
Computer Science Department/ Faculty of Electrical Engineering,
University of Science and Technology Houari Boumediene,
BP 32, El-Alia 16123 Bab-Ezzouar Algeria

Keywords: Modeling software process, intelligent agents, software process centered environment.

Abstract: This paper presents a new approach for modeling and executing software processes based on the concept of multi-agent systems. We introduce the modeling process as one of the most important goal of the agent, and we use the concept of “intelligent agent” to give more flexibility when adapting software processes to unexpected changes. This is possible thanks to the multiple capacities of the agent like autonomy and reactivity.

1 INTRODUCTION

Multiple platforms (CORBA, COM, EJB…) have been used for distributed software process development. The distribution requires 1) a performant interaction models that can be adapted to all kinds of software process interactions and 2) a dynamic software process model to support and to handle all changes and unexpected events that can happen in the environment.

The concept of the agent was used for the development of environments that focus their efforts on the execution aspects of software process models using the agent characteristics of interactions (communications, negotiations, cooperations…) and actions. However, few works was focused on the modeling aspect of software processes based agents and unfortunately, the representative capacities of the agents were less exploited.

This paper presents our approach for modeling and executing software processes based on the concept of multi-agent system. This approach allows to modelize dynamic distributed software processes. Thanks to the agent concept that has features of autonomy, reactivity and proactiveness, we use the concept of “intelligent agent” to give more flexibility when adapting software processes to unexpected changes, handling unexpected events, managing exception, etc.

Section 2 presents a brief state of the art in the domain of process software development based agents. Section 3 describes our approach for modeling and executing software processes based agents. We describe the architecture of our system through the multiple interactions between the software process agents and their internal structures. We show how, in one hand, it is possible to use the representative capacities for modeling the software processes, in parallel with action and interaction capacities; and, in an other hand, how to give more autonomy to supervise software process execution.

2 SOFTWARE PROCESS CENTERED ENVIRONMENT BASED AGENTS

This study is based on the most important software processes environments based agents: (Wang, 2000a), (Wang, 2001), (Wang, 2002), PEACE+ (Alloui, 1996) and ALLIANCE (Alloui, 2001a).
Process Centered Environment based Agents (PCE).

Main characteristics of the PCE:

- **Interaction.**
  - Communication/negotiation
  - Cooperation/Coordination

- **Distribution.**
  - Work spaces /tools/resources/developers

- **Heterogeneity.**
  - Tools/Communication languages /Message format

- **Dynamicity.**
  - (side of changements handling)
  - Dynamic handling of the changes /Possibility to insert changes during the execution of the software process model

- **Mobility of the components.**
  - Software process fragments /Software agents/Tools

- **Consistency.**
  - Of data /Of the activities

**Software process modeling.** Software process modeling is not done by the agents. The choice of these environments is due to the diversity of exploitation of their agent's capacities.

CAGIS « Cooperative Agents in Global Information Space » is a system that allows a distributed and heterogeneous work. The used software agents (system agents, local agents and interaction agents) are specialized to carry out particular tasks according to multiple services (Wang, 2000b) and (Wang, 2000c). ALLIANCE « Algebra and Logic for Interoperable Agents in Cooperative Environment » is a software framework that allows modeling, executing and supervising intensive processes (software process, workflow, automatised information system…), using three kinds of agents (the process agents, the infrastructure agents, the quantitative control agents).

PEACE+ « Process-centered Enactable and Adaptable Computer-aided Environment» is a support for software process cooperation. It uses a knowledge based formalism for the software process modeling (Alloui, 2001b). The most contribution of PEACE+ is that it presents an intentional approach for the agent interactions in order to give a rich semantic to the interactions and an efficient control through triggers.

**3 SYNTHESIS AND COMPARISON OF THE STUDIED SYSTEMS**

We resume below the main characteristics of the studied systems and the degree of their efficiencies:

The study and the comparison of the environments CAGIS, ALLIANCE and PEACE+ has lead us to notice that the different capacities of the agents were used according to their fixed objectives and to be suitable to some kind of the software development.

CAGIS is more adapted to a software development in extended sites, that are geographically distributed, and where the time is not a critical resource. Tools, development techniques and communication languages can be heterogeneous.

ALLIANCE framework is more adapted to the software development where the time is a critical resource. The communication is reduced because there is only one kind of agent to handle all the interactions in the system.

PEACE+ environment gives more independence to the developers; it can be used manually, semi-automatically or automatically. It is adapted to the software development where the ”human” aspect is very important. It gives a great autonomy for the developers to take decisions and facilitates their interactions through the interaction model previously defined.

The multiple research using the agent concept for software process development have been only interested on the process execution aspect. The software process modeling aspect didn't have so attention. If the most important part of the agent abilities like autonomy, action and interaction was exploited, the representative capacities has been neglected, particularly at the software process modeling level.

Our research work focuses both on the study and on the ability for multi-agent systems to modelize and to execute the software processes. Next section presents our approach using all these aspects.
4 OUR APPROACH FOR MODELLING AND EXECUTING SOFTWARE PROCESS BASED INTELLIGENT AGENTS

Our system is defined in terms of hierarchic software process agents. We describe below their roles, their multiple interactions and their internal architectures. We consider three kinds of agents:

**Task Agents**: the role is to execute the activities defined in the software process model. An execution report is sent to a fragment agent.

**Fragment Agents**: Each fragment agent is localized in one workspace and manages its associated software process fragments. Its goals are to 1) divide the software process fragments into sequential tasks, 2) to initialize its task agents with their associated sequential tasks and 3) to synthesize the execution results for the supervisor agent.

**Supervisor agent**: It is considered at the top of the agent hierarchy and has the global vision of the software process execution in all the workspaces; this is to ensure the global software process consistency.

The supervisor agent, considered in the top of the agent hierarchy, has a global vision of the software process execution in all the workspaces; this is to ensure the consistency of the global software process.

The tasks agents, considered at the low level, are in charge of executing the different tasks regarding the software process model. The decisions taken by a task agent are locally and prior to its workspace.

### 4.1 Internal architecture of the software process agent

After the study of the well-known internal architecture of the agents, we choose a modular architecture (Ferber, 1997), (Jennings, 2000). This architecture is suitable to our software process agent, where many parallel functions are executed. An agent architecture is modular and horizontal. Each agent is composed of six modules that work in parallel and in cooperation. Each module is responsible of many internal tasks of the agent. This is illustrated in fig 2.

![Figure 2: Internal architecture of the software process agent](image-url)
The modules of the agent software process can be classified into two categories:

1- Modules to manage the internal function of the agent: These modules are the communication module, the knowledge base management module, the resources and acquaintance module.

The functions of these modules are the same for the three kinds of the software process agent (tasks, fragments and supervisor).

2- Modules for modeling and executing the software process: The main goal is to manage the software process modeling, planning, executing and introducing process changes. These modules are: software process manager module, planning and decision module, the realization and evaluating module.

The functions of these modules are various according to the kind of the agent. For instance, the main function of the fragment agent module is to evaluate the execution results and to decide about the needed changes if necessary. Each kind of agent is responsible of a given fragment of software process and has the responsibility to manage its representation and the evolution of its execution (at the different levels of course).

5 CONCLUSION

This paper presents a new approach for modeling and executing the software process using the multi-agent concepts. A state of art, first done, on three existing environments (CAGIS, PEACE+ and ALLIANCE) shows that the priority of these systems is at the execution level of the software processes and not at the modeling level. The capacities of the agents that can handle the modeling level is being unexploited.

Our contribution is the development of a software process environment based on the concept of multi-agent system. An agent is, thus, considered as a part of a software process. In that, an agent allows not only the execution of the assigned tasks, but also, a permanent control of the modeling and the executing of the software processes. The “intelligence” aspect in our system reside in the possibility to force (anticipate) the events that can be happened and, thus, to take a pertinent decisions at a time.

The hierarchical structure designed in our system seems beneficial thanks to the cooperative aspects of the agent. The current work tends to the development of process fragmentation techniques, the evaluation techniques and the mobility of the agents.

REFERENCES


Alloui, I., et al., 2001a. A software framework for modeling, executing and flow control of intensive software processes. LLP/cesalp Laboratory ESIA Engineering School University of Savoie, France.


