APPLYING THE BDI INTELLIGENT AGENT MODEL FOR MONITORING ENTERPRISE PROJECTS

Azhari 1, Subanar 2, Retantyo Wardoyo 3, Sri Hartati 4, Edo Nurwanto 5
1,2,3,4,5 Intelligent System Research Group, Computer Sciences Gadjah Mada University, Yogyakarta
arisn@ugm.ac.id 1

ABSTRACT

In this paper, we present a part of our research on development autonomous multiagent system in monitoring enterprise projects. We have designed and developed a prototype of intelligent agent software based on BDI (beliefs-desires-intentions) model. Based on some simple data simulations of enterprise projects, our agents have showed capabilities accessing and providing the actual progress information of projects.

Keywords: Intelligent agent, BDI agent model, project progress, monitoring system.

1 INTRODUCTION

Research and applying of multiagent technology (agents based system) in information system area has developed rapidly in recent year. Intelligent information system or intelligent management support systems built based on agent technology has increased performance, and incredible capability of the application. Intelligent information systems are also better, easier, efficient, and effective compared to conventional methodology approach, such as procedural, predicate, or object oriented programming [1], [2]. An intelligent information system can be a computer artificial intelligence program that developed by using interaction, communication, coordination capabilities of intelligent agents in single agent system or multiagent system [3], [4], [5].

An interesting problem and more considerate by company management and project director is application support systems that capable to monitor company projects. Project director always requires reliable project performance information that they can trust effectively manage projects and to ensure contractual obligations. These support systems must be able to provide various data, facts, information, and project progress actually, continually, accurately, and integrally for all people involved in company projects. Both internal and external need of a company, to manage project and to improve service quality, collaborate, and communicate with stakeholders. The enterprise project monitoring software can extend the enterprise architecture company platform for collecting, analyzing, aggregating, and visualizing data that can be captured in the log and statistical models [6], [7], [8], [9].

In general, the advantage of project management software is to assist company team in collaborating, staying on track, keep within your budget and successfully completing your project. The effective project management software can handle all the complex projects of business without interruptions from unclear roles, miscommunication, convoluted tasks and lack of accountability. Organizations will increase productivity because the people and information critical to project and process success are seamlessly connected [6], [10], [11].

2 THE BDI INTELLIGENT AGENT MODEL

Definition of intelligent an agent of agent based system and multiagent system has been stated by a number of researchers in intelligent system fields. The definition still improved continually [4], [12].

In this paper, our agent definition refers to Wooldrige and Jennings [13] wrote that an agent is a computer program that has autonomous capability, can live and active in its environment. Additionally, according to [4], [14], [15] every agent usually should has one or more behaviors such as autonomous, delegated, reactive, proactive, and interaction with other agents, also making own decisions against environment change.

"An agent is a computer system that situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objective"
Regarding the theoretical foundation and the number of implemented and successfully applied systems, the most interesting and widespread agent architecture is the Belief-Desire-Intention (BDI) architecture, introduced by Bratman as a philosophical model for describing rational agents [16]. It consists of the concepts of belief, desire and intention as mental attitudes that generate human action. Beliefs capture informational attitudes, desires on motivational attitudes, while intentions on deliberative attitudes of agents. Rao and Georgeff [17] have adopted this model and transformed it into a formal theory and an execution model for software agents, based on the notion of beliefs, goals, and plans. A simple schema of the internal BDI agent model is shown in Figure 1.

\[\text{Figure 1. BDI Intelligent Agent Model}\]

According to Pokahr, et al. [18] the BDI theory of Rao and Georgeff that defines beliefs, desires, and intentions as mental attitudes can represent the possible world states in natural way. The most important is only the beliefs can be represented explicitly. The intentions of an agent are subsets of the beliefs and desires, i.e., an agent act towards some of the world states it desires to be true and believes to be possible. The desires reduced to events that handled by predefined plan templates, and intentions are represented implicitly by the runtime stack of plans to be executed.

3 THE BDI MODEL OF JADEX

Pokahr and Braubach [19] state Jadex tools can facilitate the BDI model in the context mainstream programming, by introducing beliefs, goals and plans as first class objects that can be created and manipulated inside the agent. To develop applications with Jadex, the programmer has to create two types of files: XML agent definition files (ADF) and Java classes for the plan implementations. The ADF can be seen as a type specification for a class of instantiated agents.

In Jadex, agents have beliefs, which can be any kind of Java object and are stored in a belief base. Goals represent the concrete motivations (e.g. states to be achieved) that influence an agent's behavior. To achieve its goals the agent executes plans, which are procedural recipes coded in Java. While plans represent the agent's means to act in its environment. Depending on the current situation, plans are selected in response to occurring events or goals. The selection of plans is done automatically by the system and represents one main aspect of a BDI infrastructure. In Jadex, plans consist of two parts: A plan head and a corresponding plan body. The plan head is declared the ADF whereas the plan body is realized in a concrete Java class. Therefore the plan head defines the circumstances under which the plan body is instantiated and executed.

In Figure 2, “Components of a Jadex agent” is shown how XML and Java files together define the functionality of an agent. Figure 3 shows “Jadex agent XML schema” which elements can be specified inside an agent definition file. To start an agent, first the ADF is loaded, and the agent is initialized with beliefs, goals, and plans as specified. Then the plan body will be called if any message events come into the agent.

\[\text{Figure 2. Components of a Jadex agent}\]

\[\text{Figure 3. Jadex agent XML schema}\]

Jadex BDI agents model also have mechanism to search data or facts in belief base, for example using the OQL query. According to Pokahr, et al. [13], [14] the syntax of the OQL
4 RESULT AND DISCUSSION

The application has been developed by Jadex agent tool, NetBeans 6.5 and glassfish web server v2. The application is built to help project manager or project leader in monitoring project progress by displaying the progress percentage of a project. With this application, manager can take the best decision in project management task.

User can use the project monitoring application by accessing web page “Project-Monitoring” in form of java web application. That web page contains an input text that can be used to input the project name and a “Show progress” button to submit the input to the agent. Interface that appears at the time user accessed “Project-Monitoring” web page is shown at Figure 5. The source code part of the project monitoring web page is displayed in Figure 6.

After opening the web page, user can input the project name using a textbox. Then the information progress of a project is shown by the agent. File “redirect.jsp” is used to connect the application and the jadex agent. The source code part of the file “redirect.jsp” is shown in Figure 7.

Jadex agent used to show progress of a project. Jadex Agent conducts socket communication to acquire project name sent by user. After successfully acquire the project name, Jadex Agent do searching to find the progress of the selected project. The progress then shown to user by a web page.
We develop our agent with capabilities, beliefs, goals, plans, and expressions as follow. Agent capabilities contain belief of an agent in form of beliefset. Beliefset used to store projects facts that are tasks contained in a project with progress percentage. Figure 8 shows the source code part of agent capabilities.

Belief used by agent is an exported belief that included in capabilities. Figure 9 shows the source code part to use exported belief in an agent.

The agent goal is an achieve goal and the target is to show the progress of a project. Agent goal also contain communication method using socket communication. The source code part of agent goal is shown in Figure 10.

Our Agent has two plans, such as “Find Project Plan” and “Server Plan”. The “find project plan” contains actions done by agent to show task progress of a project. One of the actions performed by agent is request to belief set to find progress of a project, which shown in Figure 11. The “server plan” contains actions to do socket communication. Figure 12 show the source code part for accepting socket communication from client.

Agent properties part of agent is used to print log messages to the console. Agent uses default configuration, but there is initial element called plan server that loaded at the agent startup. Plan server is loaded in order to acquire the data input from user through socket. The source code part of agent configuration and properties is shown in Figure 13.

Our Prototype of BDI agent model has been tested for some simple data simulation of enterprise project. The project monitoring application can display successfully by giving output in form of task progress of a project selected by user. If project name not found, application must give notification “the project is not found”. For an example, we want...
to find the progress of “proyek1”. After clicking the “Show Progress” button, “Task Progress” page will appear as shown in Figure 14a. Notifications will appear if project name not found as shown in Figure 14b and user not entered the project name as shown in Figure 14c. The source code part that is used to show the task progress is shown in Figure 15.

![Task progress interface](image1)
![Project not found](image2)
![Project null](image3)

**Figure 14. Task progress interface**

![Source code to show task progress](image4)

**Figure 15. Source code to show task progress**

5 CONCLUSIONS

Our simple prototype of intelligent BDI agent model has capability to receive the user message from the web page in order to show all project task progress. The communication process from web page to our agent is conducted by using socket communication. The progress tasks of a project are displayed to user by agent with its gShowProgress goal.

REFERENCES


