Multi-Agent Based Project Portfolio Management Approach

Lixin Zhou
School of Software and Microelectronics, Peking University, Beijing, China
zhoulx@vip.sina.com

Abstract—In project portfolio management, the sources of change, task interactions, and the necessity for distributed planning should be considered. Most existing project methods have difficulties to produce desired results in highly dynamic multi-projects environments. In this paper, we present a project portfolio management approach with multi-agent. We described the types of agents used in project portfolio management system. The mechanism of scheduling and planning of multi-project is explained in this paper. We then propose a model of project portfolio management. The software structure of project portfolio management system is presented in this paper. We described the implementation of a project portfolio management system with multi-agent, which is proven to be flexible and practical.

Index Terms—project portfolio management, agent, planning, scheduling.

I. INTRODUCTION

The fundamental objective of the project portfolio management process is to determine the optimal mix and sequencing of proposed projects to best achieve the organization's overall goals - typically expressed in terms of hard economic measures, business strategy goals, or technical strategy goals - while honoring constraints imposed by management or external real-world factors. Typical attributes of projects being analyzed in a project portfolio management process include each project's total expected cost, consumption of scarce resources (human or otherwise) expected timeline and schedule of investment, expected nature, magnitude and timing of benefits to be realized, and relationship or inter-dependencies with other projects in the portfolio.

In principle, project portfolio management [1] attempts to address issues of resource allocation, e.g., money, time, people, capacity, etc. In order for it to truly borrow concepts from the financial investment world, the portfolio of projects and hence the project portfolio management movement should be grounded in some financial objective such as increasing shareholder value, top line growth, etc. Equally important, risks must be computed in a statistically, actuarially meaningful sense. Optimizing resources and projects without these in mind fails to consider the most important resource any organization has and which is easily understood by people throughout the organization whether they be IT, finance, marketing, etc and that resource is money.

While being tied largely to IT and fairly synonymous with IT portfolio management, project portfolio management is ultimately a subset of corporate portfolio management and should be exportable/utilized by any group selecting and managing discretionary projects. However, most project portfolio management methods and tools opt for various subjective weighted scoring methods, not quantitatively rigorous methods based on options theory, modern portfolio theory, Applied Information Economics or operations research.

Project portfolio management involves ongoing analysis of the project portfolio so each investment can be monitored for its relative contribution to business goals versus other portfolio investments.

From a requirements management perspective project portfolio management can be viewed as the upper-most level of business requirements management in the company, seeking to understand the business requirements of the company and what portfolio of projects should be undertaken to achieve them. It is through portfolio management that each individual project should receive its allotted business requirements [1][2].

Software agent is an object who has autonomous actions of accomplishing specific task [3]. In multi-agent system, an agent interacts with other agents or environments to achieve its goal by communication, coordination and cooperation. Of various agent architectures which have been presented, BDI (Belief-Desire-Intention) model [4] [5] [6], is probably the most mature and has been adopted in a number of research and industrial applications. There are some works that have introduced multi-agent into the execution monitoring of business processes [7] [8] [9]. The rationale-based monitors [7] aim at addressing planning and removing the conventional assumptions in static and determinate.

In this paper, we present a project portfolio management approach with multi-agent. We described the types of agents used in project portfolio management system. The mechanism of scheduling and planning of multi-project is explained in this paper. We then propose a model of project portfolio management. The software structure of project portfolio management system is presented in this paper.

This paper is organized as follows. In Section 2, we depict types of agent. In Section 3, we present the architecture of project portfolio management system. In Section 4, we discuss the implementation of a project portfolio management system. Finally, we draw
conclusions and future works in the conclusion section.

II. TYPES OF AGENT

Intelligent agents are a paradigm of software system development. They are used in a broad and increasing variety of applications. Nowadays, researchers agree on the following definition:

The term “agent” denotes a hardware or (more usually) software-based computer system, that has the following characteristics:

- Autonomy,
- Social ability,
- Reactivity,
- Pro-activeness.

In the same manner that there are several languages to implement agents, there are also different levels of complexity of this implementation. Such complexity depends on the task that agents have to carry out and on the environment surrounding them. The following is classification of agent architectures:

- Simple reflex agents.
- Model-based reflex agents.
- Model-based, goal-based agents.
- Utility-based agents.
- Learning agents.

III. SYSTEM ARCHITECTURE

We adopt multi-agent system as information infrastructure to support project management in a highly distributed environment. A project has five important targets: scope, schedule, cost, quality and resources. Scope, schedule, cost, quality and resources needed in a project are represented by an agent. These agents reside at the site of the project which team members own the resource or implement the activities. The portfolio and each project are represented by an agent. Functions of project management are taken by service agents. The agents in project portfolio are shown in Fig. 1.

Several kinds of agent are used in project portfolio management system, including simple reflex agents, model-based reflex agents, model-based, goal-based agents, utility-based agents, and learning agents. The relations of different kinds of agent are shown in Fig. 3.

A belief-desire-intention-capability (BDIC) architecture includes and uses an explicit representation for an agent’s beliefs, desires, intentions and capabilities. Capability component determines agent’s types and monitoring functions.

Simple reflex agents take charge of collecting or obtaining data for a project. Goal-based agents take charge of project planning. Utility-based agents take charge of portfolio planning. Learning agent response for obtaining historical information and improve the next in project portfolio management. There is a whole agent for coordinating the multi-agents.

IV. IMPLEMENTATION

A project portfolio management system have been successfully implemented and seamlessly integrated. Agents act as intelligent software unit. Project portfolio management system runs on a J2EE application server, namely Weblogic, and adds intelligence in the sense of intention-based monitoring behaviors. Under the J2EE framework, session beans are components containing business logics associated with a particular client session or task. We adopts a correlative work, called the contract-based interlayer, to integrate the layer of system software-related functions with the layer of business-related functions. With the contract-based interlayer, Multi-Agent System can send out alerts to business partners’ services. Multi-Agent System cooperates with Web Daemon in providing access control functions.

Multi-Agent System targets at providing a mechanism for business process management in a web services environment. Multi-Agent System is loosely coupled with other applications in architecture in the sense that it
can be plugged into any potential systems, including e-commerce, web services and web-based applications. There is no implementary obstacle for these systems if only they hold out the specification of web services.

### A. Agent Type in a Project

In a project, scope agent, schedule agent, cost agent, quality agent and resources agent are defined as goal-based agent. A goal-based intelligent agent model is defined in [10], is presented in Figure 4.

<table>
<thead>
<tr>
<th>Interface</th>
<th>Application Functions and Logic</th>
<th>Agent Functions and Logic</th>
<th>Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goal Model – Execution Plan</td>
<td>Schedule Management Functions and Logic</td>
<td>Schedule Goal Model – Execution Plan</td>
<td>Environment</td>
</tr>
</tbody>
</table>

Fig. 4 The goal-based intelligent agent model

The composite state goal model defines intelligent behavior of an agent. States represent the goal, sub goals and transit states. Transitions represent the dynamic behaviors. The tasks being fulfilled in each transition consist of functionalities of the agent application, which include agent basic functions and application specific functions.[10]

In this model, the environment layer defines the agent running environment including system environment (system architecture, operating system, network, knowledge base, etc), communication mechanism (communication method, communication language, etc) and agent development environment (development language, agent framework, etc).

The goal-model layer defines the goal model of agent. The goal model is derived based on the business processes, work flow and business logic of the application. The goal model will become the execution plan of agent when the agent is running.

The function layer includes application functions and logic component and agent function and logic component. Application functions and logic component consists of application specific functions and business logic. They are application dependent. Agent functions and logic component consists of agent basic functions and working logic. They are application independent. The transition objects will invoke both application functions and agent functions during task fulfillment according to the goal model.

The interface layer facilitates the agent to interact with users, other agents and the running environment. The agent communicates through the interface layer either by application functions or agent own functions,[10]

We defined scope agent as a goal-based agent which is presented in Fig. 5. A scope agent includes specific parts for scope management in a project, such as Scope Management Interface, Scope Management Functions and Logic and Scope Goal Model – Execution Plan.

<table>
<thead>
<tr>
<th>Scope Management Interface</th>
<th>Schedule Management Functions and Logic</th>
<th>Agent Functions and Logic</th>
<th>Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scope Model – Execution Plan</td>
<td>Schedule Goal Model – Execution Plan</td>
<td>Environment</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 5 scope agent model

We defined schedule agent as a goal-based agent which is presented in Fig. 6.

We defined cost agent as a goal-based agent which is presented in Fig. 7.

We defined quality agent as a goal-based agent which is presented in Fig. 8.

We defined resource agent as a goal-based agent which is presented in Fig. 9.

<table>
<thead>
<tr>
<th>Schedule Management Interface</th>
<th>Cost Management Functions and Logic</th>
<th>Agent Functions and Logic</th>
<th>Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schedule Goal Model – Execution Plan</td>
<td>Cost Goal Model – Execution Plan</td>
<td>Environment</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 6 schedule agent model

<table>
<thead>
<tr>
<th>Cost Management Interface</th>
<th>Quality Management Functions and Logic</th>
<th>Agent Functions and Logic</th>
<th>Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost Goal Model – Execution Plan</td>
<td>Quality Goal Model – Execution Plan</td>
<td>Environment</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 7 cost agent model

<table>
<thead>
<tr>
<th>Quality Management Interface</th>
<th>Resource Management Functions and Logic</th>
<th>Agent Functions and Logic</th>
<th>Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality Goal Model – Execution Plan</td>
<td>Resource Goal Model – Execution Plan</td>
<td>Environment</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 8 Quality agent model

<table>
<thead>
<tr>
<th>Resource Management Interface</th>
<th>Utility Model – Execution Plan</th>
<th>Agent Functions and Logic</th>
<th>Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource Goal Model – Execution Plan</td>
<td>Utility function</td>
<td>degree of agent’s happiness</td>
<td>Environment</td>
</tr>
</tbody>
</table>

Fig. 9 Resource agent model

### B. Project Agent and Portfolio Agent

In a project portfolio management system, project agent and portfolio agent are defined as utility-based agent. A utility-based intelligent agent model is presented in Figure 10.

<table>
<thead>
<tr>
<th>Interface</th>
<th>Application Functions and Logic</th>
<th>Agent Functions and Logic</th>
<th>Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utility Model – Execution Plan</td>
<td>utility function</td>
<td>degree of agent’s happiness</td>
<td>Environment</td>
</tr>
</tbody>
</table>

Fig. 10 Utility-based agent model

In Goal-based agents only distinguish between goal states and non-goal states. It is possible to define a measure of how desirable a particular state is. This measure can be obtained through the use of a utility
function which maps a state to a measure of the utility of the state. As a result, in order to improve the quality of agent behavior, the utility function maps agent’s state (or a sequence of states) in the model of the world, onto a real number describing the associated degree of agent’s happiness.

The goals of a project are not simple goals. In order to achieve success, we must make tradeoff in conflicting goals in the process of project implementation. A project utility function should be defined for conflicting goals in a project. So, it is better to define a project agent as a utility-based agent. We defined a project agent as a utility-based agent which is presented in Fig. 11.

The goals of project portfolio are not simple goals. In order to achieve success, we must make tradeoff in conflicting goals in the process of project portfolio implementation. A portfolio utility function should be defined for conflicting goals in a portfolio. So, it is better to define a portfolio agent as a utility-based agent. We defined a project portfolio agent as a utility-based agent which is presented in Fig. 12.

\[
\begin{align*}
\text{Project Interface} & \quad \text{Agent Functions and Logic} \\
\text{Project Management Functions and Logic} & \quad \text{Project utility function} \\
\text{Project Utility Model – Execution Plan} & \quad \text{degree of project agent’s happiness} \\
\text{Environment} & \\
\end{align*}
\]

Fig. 11 project agent model

\[
\begin{align*}
\text{Project Portfolio Interface} & \quad \text{Agent Functions and Logic} \\
\text{Project portfolio management Functions and Logic} & \quad \text{Portfolio utility function} \\
\text{Portfolio Utility Model – Execution Plan} & \quad \text{degree of Portfolio agent’s happiness} \\
\text{Environment} & \\
\end{align*}
\]

Fig. 12 project portfolio agent model

C. Cooperation and Communication Model

In the process of management of a project, scope agent, schedule agent, cost agent, quality agent and resources agent should cooperate to achieve the goals of a project. Utility function of a project can be defined as a set of goals including scope goal, schedule goal, cost goal, quality goal and resource goal.

Utility function of a project is presented in the following formula (1).

\[
\begin{align*}
\text{Utility}_{\text{project}} & := F(S, S T, Q, C, R) \\
S & := \text{ScopeGoals} \\
S_T & := \text{ScheduleGoals} \\
Q & := \text{QualityGoals} \\
C & := \text{CostGoals} \\
R & := \text{ResourceGoals} \\
\end{align*}
\]

V. CONCLUSION

In this paper, we have presented our work Multi-Agent System, a multi-agent system for project portfolio management. Multi-Agent System can be integrated with applications in the enterprise and web services provided by cross-organizational business partners in architecture. In the future, we will focus on developing richer monitoring capabilities and introducing Multi-Agent System in more domains, such as Semantic Web Services. In our view, introducing semantics into Multi-Agent System is a promising approach to make functionalities of automatic process possible.

REFERENCES