Engineering Applications of Artificial Intelligence 25 (2012) 837–845

Contents lists available at SciVerse ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

An agent-oriented approach to process partition and planning in migrating workflow systems

Jie Cheng [a,b,]*, Guangzhou Zeng [a]

[a] School of Computer Science and Technology, Shandong University, Jinan 250101, China
[b] School of Mechanical, Electrical & Information Engineering, Shandong University at Weihai, Weihai 264209, China

ARTICLE INFO

Article history:
Received 13 March 2011
Received in revised form
1 August 2011
Accepted 28 September 2011
Available online 19 October 2011

Keywords:
Migrating workflow
Workflow partition
Process planning
Markov decision process
QoS objective

ABSTRACT

Mobile agents act as the task executors in migrating workflow system. As the size of workflow is increased by including many tasks and branches, multiple mobile agents should be used where each agent is responsible for a branch of the workflow process and fulfills the workflow goal by cooperating with its partners. In this case, the workflow process needs to be partitioned into a set of sub-processes before execution so that each sub-process can be assigned to one mobile agent. This paper proposes a structured process partition approach that includes process structure partition and Quality of Service(QoS) objective partition. The former partitions a structured process into a set of sub-processes with dominant relations and each sub-process consists of a sequence of tasks. The latter distributes QoS objectives, such as the expected budget or deadline of the whole workflow, over all of the sub-processes. In addition, a sub-process execution planning algorithm with QoS objective based on Markov Decision Process (MDP) is also put forward in this paper. The experiment results show that the effects of workflow partition method and MDP based process planning method are sound.

1. Introduction

By definition of Workflow Management Coalition (WfMC), workflow is an automation of a business process, in whole or parts, where documents, information or tasks are passed from one participant to another to be processed, according to a set of procedural rules (Workflow Management Coalition, 1999). The migrating workflow is a new model for workflow management by adopting mobile agent technology (Cichocki and Rusinkiewicz, 2004), in which the mobile agent migrates from one site of network to another in order to achieve workflow goal by using site services. The behavior of mobile agent in the migrating workflow model is similar to people working in moving manner, such as buyers move from shop to shop, information searchers move from library to library, etc.

Traditionally, the workflow management system is constructed with one or more centralized workflow engines and relied on remote procedure call (RPC) principle (Workflow Management Coalition, 1999). However, the RPC depends on the reliability of network connection and lacks the adaptability to network environment. The mobile agent computing paradigm changes RPC to local procedure call (LPC), that is, a mobile agent moves among sites of network and utilizes the local services to achieve its goal. Except for moving time, a mobile agent does not require any reliability of network connection and can alternate its workplace when the expected site is down or expected service is changed. Therefore, adopting mobile agent technology can bring more flexibility to workflow management.

Like many applications of mobile agent technology (Loke and Zaslavsky, 2001; Foster et al., 1999; Merz et al., 1997; Meng et al., 2000), in the migrating workflow system, the whole process of workflow can be executed by a single mobile agent. But when the size of workflow is increased, i.e., comprising many tasks and branches, the workflow completion time will be prolonged because all of the parallel tasks have to be executed serially by one mobile agent. In such a case, multiple mobile agent mechanism should be adopted to support execution of parallel tasks in order to improve the workflow system performance (Feng et al., 2007). In such workflow systems, each mobile agent will be responsible for a part of workflow process and fulfill the workflow goal by cooperating with its partners. To this end, the workflow process needs to be partitioned into a set of sub-processes before execution so that each sub-process can be assigned to a mobile agent.

As indicated in the literature (Tan and Fan, 2007), process partition is a primary step for distributed workflow management model and some methods have been obtained so far. We will introduce these approaches as the related works in Section 6.

* Corresponding author at: School of Computer Science and Technology, Shandong University, Jinan 250101, China. Tel.: +86 13371173456; fax:+86 631 5688338.
E-mail addresses: sdchjie@163.com, chjie@sdu.edu.cn (J. Cheng), gzzeng@sdu.edu.cn (G. Zeng).

0952-1976/$ - see front matter
doi:10.1016/j.engappai.2011.09.027

Considering the decentralized, loosely coupled and autonomous nature of mobile agent computing paradigm, the mechanism of process partition for migrating workflow is quite different from those methods appropriate for the WfMC model. First, there is no central workflow engine to coordinate the workflow services and mobile agents in a migrating workflow system. Instead, the dominant relations among mobile agents, such as synchronization of tasks and coordination of service conflicts, have to be handled by the mobile agents themselves. This must be taken into account in the workflow partition. Second, the looped task and logical or-relative tasks, e.g., or-Split and or-Join need not be considered, because a looped task can be executed by one mobile agent in the same workplace, it does not require mobile agent moving; a group of logical or-relative tasks can be treated as a single task with redundancy and left choice to mobile agent. In other words, the migrating workflow partition method must be mobile agent oriented. Finally, QoS objective is also a major aspect to evaluate the performance of workflow applications. Since the price and time of service may be different among service sites even if they provide same services, thus execution time and cost are usually the main constraints to be considered.

Based on above considerations, this paper proposes an agent-oriented workflow partition approach that partitions the workflow process into a set of sub-processes; each of them consists of a sequence of tasks and will be mapped into a mobile agent. We denote the dominant relations among sub-processes as a tree structure, where nodes represent sub-processes, and upper sub-processes dominate the lower ones in cooperative semantics. After sub-processes are assigned to mobile agents, the dominant relations among the sub-processes will become the cooperative relations among the mobile agents. Based on the structure partition tree, we present two kinds of QoS objective partition method named as minimal cost within a deadline (MCD) and minimal time with a budget (MTB). Due to the sequential structure of sub-processes, a task execution planning method is also proposed by modeling the sequential sub-process as a Markov Decision Process (MDP). The working path for a mobile agent is constructed based on the MDP sub-process planning with QoS objective.

The remainder of this paper is organized as follows. Section 2 gives an overview of migrating workflow framework. The workflow process partition approach and planning methods are detailed in Sections 3 and 4, respectively. Section 5 illustrates the experimental results. In Section 6, we introduce the related works and make some discussion. Section 7 concludes the paper and gives the directions for future work.

2. Migrating workflow framework

In a previous work (Zeng and Dang, 2003), we proposed a migrating workflow framework based on the mobile agent computing paradigm and called a mobile agent as a migrating instance (MI). The migrating workflow framework comprises three autonomously and loosely coupled components:

1. A migrating workflow engine used by a workflow manager to define workflow, create, launch and monitor MIs but not coordinate their work.
2. A set of MIs that will fulfill the tasks on behalf of workflow goal using local services of workplace.
3. A set of workplaces that provide run time environment and local services to MIs.

In order to make the semantics of process partition and planning with an expected QoS objective more clearly, the following definitions have been improved in this paper.

Definition 2.1. A **migrating instance** constructed with mobile agent paradigm is a movable proxy of workflow manager on behalf of a workflow goal, denoted as $MI=(MI_{id}, Tlist, Corr, WD, WPath, Code)$, where

- MI_{id} is the unique identifier of the *MI*, which can be recognized by its partners and workplaces.
- *Tlist* is a sequence of tasks with QoS objective such as the limit of payment or execution period.
- *Corr* is a set of cooperative relations among the *MI* and its partners.
- *WD* is set of data used or produced by the *MI* during the execution period.
- *MPath* is a sequence of workplaces for executing *Tlist*.
- *Code* is a set of programmed functions to support MI to schedule tasks, cooperate with its partners, interact with its owner, guard safe itself, etc.

Definition 2.2. A **workplace** is a site of network provided by a member of workflow coalitions, defined as $wp=(wp_{id}, wServ, wState, wMach)$, where

- wp_{id} *is* the unique identifier of the *wp*.
- *wServ* is a set of workflow services, each service is denoted as $serv=(ability, price, time)$, in which *ability* denotes the service ability represented by a number of functions with computing complexity, *price* and *time* respectively express the service cost and time spent on a unit task. The two-tuple (*price*, *time*) reflects the QoS of the *wp*.
- *wState* denotes the current state of workplace while MI is expecting to arrive on it, such as normal, failure, busy or service changed.
- *wMach* is the work machine of workplace to support runtime services and workflow services when MI has been arrived on it, the runtime services usually include creation of run time environment, moving in and moving out, communication, safe guard, etc.

For convenience, we use notation $*.\#$ to express # is a part of $*$ throughout the rest of this paper. For example, *wp.wServ* denotes the set of service of the workplace *wp*.

Given a task t, we call that a workplace *wp* is available to task t if there exists a service $serv \in wp.wServ$ that can meet the matching between task t and the service *serv*. That is, the workplace *wp* has ability to execute task t. In this paper, we denote the matched service of workplace *wp* for task t as S^t_{wp}. According to the definition of workplace, the cost consumed on task t by S^t_{wp} is in proportion to the size of task t and $S^t_{wp}.price$, and the time spent on task t by S^t_{wp} is in proportion to the size of task t and $S^t_{wp}.time$.

Let *WP* be the set of all workplaces in the migrating workflow system. Given a task t, we define $WP(t) \subset WP$ as a sub-set of *WP*, in which all of workplaces are available to task t but may have different QoS. That is, $WP(t)$ is a redundant workplace set in terms of service ability for task t.

The main steps for a migrating workflow execution are as follows:

1. Define a workflow process with a QoS objective.
2. Partition the workflow process into a set of sub-processes, each sub-process consists of a set of tasks.
3. Distribute the expected QoS objective over the sub-processes.
4. Plan an execution path *MPath* for each sub-process in terms of its sub-QoS objective. If the sub-QoS objective cannot be reached based on available services, it will be adjusted and