ADAPTIVE USE OF TASK ASSIGNMENT MODELS IN TEAM-BASED MOBILE BUSINESS PROCESSES

Habin Lee
Brunel University

John Shepherdson
British Telecommunications plc.

Hyung Jun Ahn
Waikato University

Abstract

Most mobile business processes are executed under uncertain and dynamic working environments. This makes the traditional centralized approach for the management of mobile tasks inappropriate to respond to the changes in working environment quickly as collecting the changing information from geographically distributed workforces in real time is expensive if not impossible. This raises the need of a distributed approach in the management of mobile tasks. This paper proposes a distributed architecture for team-based coordination support for mobile task management. In this architecture, tasks are managed via peer-to-peer style coordination between team members who have better understanding on the changing working environment than a centralised system. The novelty of the design of the architecture is explained by applying it to a real business process in the UK.

Keywords: *Business process, Mobile computing, Agent, Task Assignment Model.*

1 INTRODUCTION

Today’s competitive market environment requires mobile service providers to meet ever-increasing high customer expectation. From this perspective, the management of mobile business processes (mBPs) often put a high priority on meeting time commitments with customers. This requires today’s workflow management systems (WFMSs) to have more enhanced mechanisms in assigning tasks to mobile workers overcoming uncertainties such as traffic jam and appearance of urgent tasks, and also taking into account various task characteristics such as deadline, required skills and current geographic location of workers. However, today’s centralised WFMSs usually have difficulty in collecting all the changing information in the field in real-time, leading to incorrect schedules based on outdated or inaccurate information that may result in inefficient work coordination or expensive human intervention to correct the schedules (Lessaint et al., 2000).

One of the recent trends to overcome the problem is team-based approach wherein tasks are assigned to a team not to individual workers and the team decides who completes which tasks via a task assignment model (TAM) based on their local knowledge (Lee et al, 2007).

However, the dynamic nature of mobile working environments may require the teams to use multiple TAMs interchangeably to meet customer commitments. For example, in a distributed TAM (DTAM) wherein a team leader plays a crucial role in the distribution of team tasks to team members, the temporary absence of the team leader would be critical in the management of team tasks. In this case, the team leader may want to use a centralised TAM (CTAM) for a while or choose to delegate his team-leader role to one of his team members. The team leader may also want to use multiple TAMs for different task instances so that tasks that require special care can be assigned to a worker via a special TAM while other tasks might be handled via a normal TAM. This requires an information system for team-based task management to support a flexible role management, dynamic switching of TAMs and selective use of multiple TAMs for different task instances, as well as basic team configuration and interaction among the team members.

This paper aims to propose a novel design of information system architecture called TeamWork that meets the above requirements for team-based task management in the mobile business processes context. TeamWork is applied to a real mobile business process to show its usefulness.

This paper is organized as follows. Section 2 reviews related work and section 3 presents TeamWork. Section 4 shows an illustrative example and section 5 summarises this paper and presents future research directions.

2 LITERATURE REVIEW

Many DTAMs have been proposed in the literature mainly to increase the flexibility of task assignment to a group of process actors. Contract-Net Protocol (CNP) is one of the most popular ones as a DTAM (Davis and Smith, 1983). In a CNP, there are two types of role: an initiator and participants. At the beginning, the initiator sends out a Call-For-Proposal (CFP). Then each participant reviews the CFP and places a bid if it thinks it can make a profit from it. After predefined time duration, the initiator evaluates all the bids received from the participants and it selects one or more best bids and rejects the others. If we apply CNP to a task assignment problem, then a CFP may contain task information that should be executed by one or more participants. Participants, mobile workers in this case, will then submit bids which describe the start time of the task, the travel costs, the level of their skills, and other relevant information. The initiator, then, may choose one or more workers who can start the task and meet customer requirement best. Since the release of CNP, several negotiation protocol designs have been also proposed in the literature (Strobel, 2000; Varian 2001; Davis and Smith, 1983). Market-based mechanism also has been widely used as a negotiation protocol (Geppert et al., 1998).

DTAMs show several advantages over centralised TAMs (CTAMs). First, a DTAM is a very powerful tool to resolve a large number of scheduling conflicts (Graves, 1981). Second, DTAMs using auction-based mechanisms can be robust in resolving conflicts, are efficient in allocating scarce resources such as heating power in a building or ATM network bandwidth, and have an adaptive design (Tan and Hacker, 1999). It has been also suggested by Tan and Harker (1999) that DTAMs show better performance in workflow coordination over CTAMs where information technology is cheap, processing time is relatively long, and the pool of agents is not large. Last but not least, DTAMs can be a very useful tool when managers want to implement incentive mechanisms for their workforces. For example, if mobile workers receive bonus based on the points they earn by executing more tasks, DTAM can provide an incentive mechanism easily while CTAM cannot.

While DTAMs have been mostly proposed and discussed in academia, CTAMs have been more widely used in reality. For example, BT has been used a centralised workforce scheduling system since 1997 (Lessant et al. 2000) for its mobile field workers for install and maintenance of telecommunication services. The low utilization of DTAMs can be attributed to the immaturity of infrastructure for the implementation of DTAMs. For example, multi-agent systems or peer-to-peer computing technology that are supposed to be most ideal for implementing DTAMs are still yet to see commercial success.

Dynamic selection of task assignment policies has been proposed in the distributed computing systems field (Harchol-Balter, 2002). The study aimed at distributing computer work loads to distributed computing servers, investigating the relationship between task size variability and specific task assignment policies.

3 TeamWork: AN ARCHITECTURE FOR TEAM-BAED TASK MANAGEMENT

TeamWork is a distributed architecture that can be used by a team of workers to mange team tasks. The core of TeamWork is a task management model that describes how a set of team tasks can be assigned or re-assigned to team members via cooperation. As a result, TeamWork consists of a task management model, participating role model and their interaction models. In TeamWork, team members can determine a task assignment model, switch to another or change rules of the model flexibly.

3.1 Team-based Task Management Model

A team-based task management model (TTMM) defines the structure and coordination processes for the management of team tasks. Team structure describes which roles are involved in the management and a coordination-process the flow of the interactions among the defined team roles. Meta coordination process defines the sequences among the coordination processes in the task management. A TTMM provides all the participating agents in target system with contextual information on the team working. For the convenience of interpretation of the agents, this paper adopts XML as the basic description language of TTMM. A part of the DTD of TTMM and a diagrammatic model of an example task assignment model (TAM) are shown in Figure 1.

```
<!ELEMENT ttmm (team, adj-team+, tam+, def-tam, exceptions+)>
<!ELEMENT team (id, name, area)>              <!ELEMENT adj-team (team-id+)>
<!ELEMENT tam (tlm, role+, service+)>         <!ELEMENT tlm (ini-stat, status+, trans+)>
<!ELEMENT trans(pre-sta, service, post-sta)>  <!ELEMENT service (ccom+, seq+)>
<!ELEMENT seq (pre, post, cond+)>             <!ELEMENT exceptions (cond+, tam-id)>
<!ELEMENT cond (attr, value)>
```

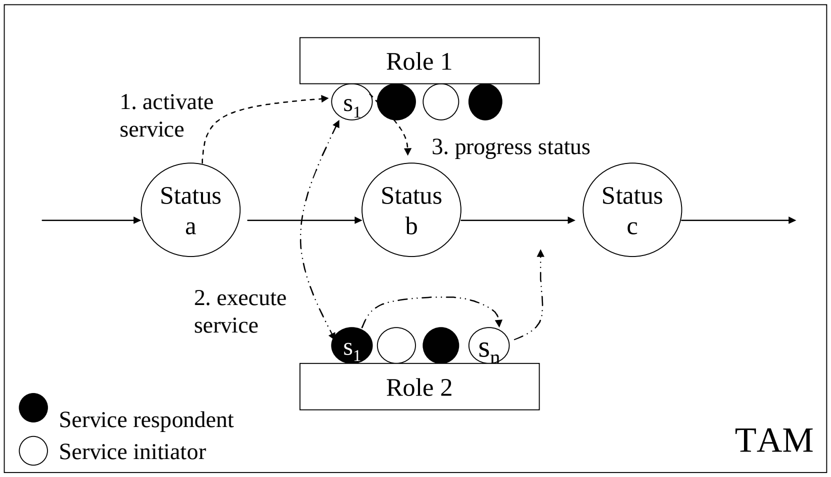

Figure 1 DTD of a TTMM and a diagramatic model of a task assignment model (TAM)

A *ttmm* is described with team information such as *team id*, *name* and responsible geographical *area*; adjacent teams (*adj-team*); task assignment model (*tam*) that specifies the task delivery path through the roles and related rules defined in the team; default tam (*def-tam*) that is the default tam that is used when a team starts a working day; and exceptional rules that enforce tasks having specific attribute values are assigned by predefined *tam*s rather than default tam. A *tam* is described with task lifecycle model (tlm), participating roles and allowed services that are necessary for the management of tasks. A *tlm* defines the possible transition paths of task status and described with initial status (*ini-stat*), a set of possible task status (*status*) and a set of transition (*trans*) that defines which service can be executed when a task reached to a status and make it progress into which task status. A *tlm* provides useful information in determining which services should be accessible by a team member or the leader. For example, in Figure 1, if the status of a task reaches "Status a", the model informs that any team members who are in charge of role "role 1" can start a coordination service "s1". Once any instance of role "role 1" starts the coordination service "s_1" this may trigger another service "s_n". Once the coordination service is completed, it may update the task status into "status b". A cooperative service (*service*) is described with cooperative service components (*ccom*) and their sequences (*seq*). The sequence of *ccom*s is defined by a priori ccom (*pre*), a posterior ccom (*post*) and conditions (cond) for the transition from *pre* to *post*.

Exceptional rules are useful tool for team leaders who want to use multiple TAMs in different situations interchangeably. Exceptional rules specify any tasks having predefined attribute values should be assigned via specific TAM rather than the default TAM. As a result, a team can have multiple operating TAMs at a time for different task instances.

3.2 TeamWork Architecture

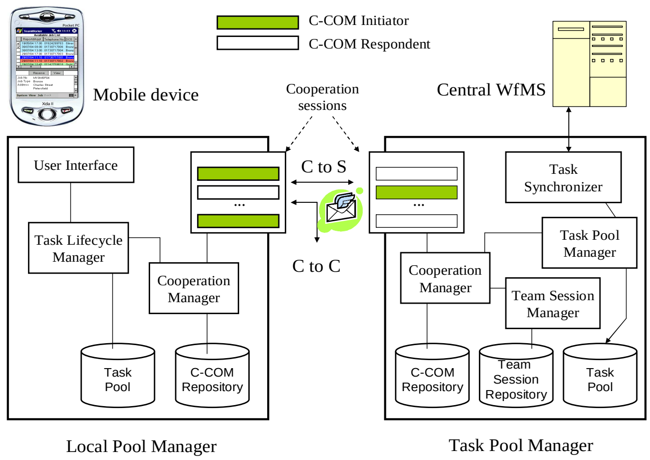

Figure 2 Components and their interactions in TeamWork

Two major components of TeamWork are task pool manager (TPM) and local pool manager (LPM). TPM is normally located in the server side and responsible for the management of all the tasks and team members for a mobile business process. On the other hand, LPM is normally located in a mobile device and represents a process actor's view.

The major roles of TPM are to collect tasks from an external source for the team, initiate a task assignment model, notify any new or urgent tasks to the team and update any changes on the attributes of the tasks such as task owner or status. At its launching time, TPM reads a TTMM provided by an administrator and initialises team settings and the task pool. Task synchronizer is a sub component of TPM that connects to an external task source such as a WFMS and collects all the tasks for the team. It has been designed to use a standard interface defined by the WfMC (workflow management coalition) so that it can be integrated with heterogeneous WFMSs. Task synchronizer passes any tasks collected to task pool manager that checks the attributes of the tasks and attach a TAM to each task according to the default TAM value and exceptional rules in the TTMM. During this process, the task pool manager contacts team session manager to have the current default TAM of the team and any changes on the exceptional rules. The team session manager maintains team session repository that keeps all the history information with regard to the team configuration. Once each task is attached with a TAM, the task pool manager executes the TAM so that the task is assigned to a worker. For this purpose, the task pool manager contacts the team session manager to get the contact information of the current team leader or any delegated team member in the absence of the team leader. After that, the execution of the TAM is managed by coordination manager. In TeamWork, all coordination among team members including the team leader is performed via C-COMs (Lee et al., 2003). A C-COM is a software component that automates coordination processes among organizational roles. It consists of two or more role components that implement the logics of the roles participating in the coordination process. For example, a C-COM implementing the auction process for a distributed task assignment will consist of two role components: auction manager and auctioneer. The auction manager role component is installed within the C-COM repository of the TPM while auctioneer role component is installed within the C-COM repository of the LPM. The coordination manager gets an instance of the auction manger role component and put into the cooperation session that executes the role component. The execution of the role component triggers a series of message exchanges with counter role components (auctioneer) in mobile devices of the workers.

An LPM is responsible for the interactions with a TPM for the collaborative management of team tasks. At its launching time, it reads the profile of the worker which specifies the worker's team profile and the roles of the worker. Based on the worker profile, it initialises C-COM repository and a task pool that contains tasks that the worker is interested in. Initialisation of C-COM repository includes selecting C-COMs based on the user role and installing corresponding responding role components of the selected C-COMs. Tasks in the task pool are largely classified into two categories: tasks already assigned to the worker and tasks in the middle of an assignment process. The worker has different views on the two types of the task on the user interface. The tasks in the task pool can have different statuses according to the task lifecycle models they are attached to. The tasks and status information are displayed on the mobile device via a user interface. Any messages from TPM are handled by the cooperation manager within the LPM. The cooperation manager identifies the relevant task of the cooperation message, and checks the progress of the cooperation process for the assignment of the task. If the message is about new task which need to be assigned to a worker, then task lifecycle manager is informed with the task information along with the TAM used so that it can create a new TLM history. Once the TLM is settled down (that is, the task is assigned either to the worker or others), then the task is either transferred into his personal task pool or removed from the pool. The message exchanges can happen between an LPM and a TPM (C to S) or two or more LPMs (C to C) depending on the used TAM.

3.3 Implementation

TeamWork has been implemented as a part of mPower project (Lee et al., 2007) that aims to develop an application framework for agent-based workflow systems for mobile business processes. The major players of the framework are Personal Assistant agents that are installed on mobile devices to support mobile workers and Information Agents that collects tasks or relevant knowledge for the execution of the tasks. Local Pool Manager has been implemented within the Personal Assistant agent for the cooperative task management and Task Pool Manager is one of the core components of an Information Agent. Jade (Bellifemine et al., 2003) has been used as the implementation platform of the agents. The user interface on mobile devices has been implemented using SWT (Software Widget Toolkit) to enhance the performance of the overall system.

4 AN ILLUSTRATIVE EXAMPLE

B Company operates about 24,000 field forces to install and maintain telecommunications networks in the UK. The field forces are working in their patches they are supposed to work in. The nature of their tasks is diverse depending on the requirements of the customers for installation tasks and the cause of the errors on the network for repair tasks. A simple installation task may require an engineer while more complicated tasks a group of engineers having different types of skill. Currently, the company is operating a centralised WFMS that assigns tasks to the field forces considering various factors such as deadline of the task completion, travelling time, the dependencies between tasks, skills required and so on (Lessaint et al., 2000). However, the centralised WFMSs are limited in considering all the significant constraints and the dynamic nature of the working environment of the field forces makes accurate scheduling very difficult. As a result, human intervention is often inevitable to make corrections on some part of the schedules made by the WFMS.

The company is now considering a team-based working wherein a team, not individual worker, has responsibility in completing tasks assigned to its patches. A team leader has authority in assigning the team tasks to its team members. Each team can work under three different modes. Firstly, they can use existing centralised WFMS. The use of the centralised WFMS can be done in direct-gateway or indirect-gateway-mode. In indirect-gateway-mode, all the tasks are assigned to the team members by the WFMS but the team leader should publish the assignments to make the tasks visible by the team members. Also, the team leader can modify the assignment based on his local knowledge before the tasks are published. In direct-gateway-mode, the assigned tasks by the WFMS are made visible by the team members without the interference (publication) of the team leader. Secondly, the teams can use the first-come-take rule for the team task assignments. In this mode, team tasks are put into the team task pool without any provisional assignment by the centralised WFMS. Every team member has the equal view on the task pool and can reserve any tasks they want to perform based on the first-come-take rule. The team leader can give restriction to this mode, for instance, setting the maximum number of tasks a member can reserve in a day. The team leader should be able to change from one mode into another based on the team's situation. For example, a team leader may become unavailable for a short term. In this case, s/he can either delegate his team leader role to one of the team members or change the working mode into first-come-taker mode. Furthermore, the team leader can use the two or three modes at the same time for different task instances. For example, s/he can configure the team operation so that tasks that have been retained by a team member are delivered to the team member next day via direct-gateway-mode while other tasks are assigned to team members via indirect-gateway-mode.

If a team member cannot complete an assigned task due to any reasons, s/he can either trade the task to other team members via task-trading services that use peer-to-peer style negotiation (in direct or indirect-gateway-mode case) or remove the reservation and put the task back to the team task pool (in first-come-taker mode case). There are two types of task trading services: mini-trading and maxi-trading. Mini-trading service uses horizontal peer-to-peer style negotiation to find new owner of a task. It first finds a list of candidates for the new owner of the task based on their current locations, availability, the task location and required skills. Then it sends an offer to each of the candidates. Anybody who accepts the offer first becomes the new owner of the task. If a mini-trading cannot find a new owner, then the team member can execute a maxi-trade service which directly contacts team leader to ask to reassign the task. If the team leader cannot find any candidate after checking the schedules of the all team members, then s/he can start mini-trading with the team leaders in adjacent teams.

A part of the TTMM for the above requirements is shown in Figure 3 (a). In the figure, <tam id=indirect-gateway-mode> tag specifies the details of the indirect-gateway-mode TAM. The inner tag <service> specifies that "job trading" service is provided via "mini-trading" and "maxi-trading" ccoms and "mini-trading" should be executed before "maxi-trading" and the later only can be executed if and only if the output of the former ccom is "failed". <exception> tag is also indicating that any task instances that have status as "retained" should be dispatched to team members via direct-gateway-mode.

Figure 3 (b) shows the screenshots of the implementation based on the specification. The first screen shows the coordination status window which displays all the incoming or outgoing coordination items with regard to team tasks management. In the figure, the incoming coordination window has a mini-trading offer from a colleague (Jamie Jones) and it should be responded within about nine and half minutes. If the user selects the item and clicks the "view" button, the systems displays the detail of the trading offer including task information and available choices for the offer (accept, reject or return).

```
<ttmm>
<team id=suffolk>
<default-tam id=indirect-gateway-mode>
<tam id=indirect-gateway-mode>
<role id=TL \>
…
<service name="Job trading">
<ccom id=mini-trade role=ini-resp \>
<ccom id=maxi-trade role=ini-resp \>
<seq><pre>mini-trade</pre>
<post>maxi-trade</post>
<cond><attr>output</attr><val>failed
</val></cond></ service></ tam>
<tam id=first-come-take-mode>
   …
<rule><attribute id=max-no-daily-
reserved-job />
<value>10</value></rule></tam>
<exception>
<cond><attr>status</attr><val>retained
</val></cond><tam-id>direct-gateway-
mode</tam-id></exception>
…
</ttmm>
```

(a)

(b)

Figure 3 (a) The team-based task management policy of B Company (b) the screenshots of the application based on the policy

5 CONCLUSIONS

Despite the recent trends toward empowering and team-based working, there has been only limited support from information systems so far. This paper considers the key reason for this as lack of support for practical operation of team-based working under dynamic working environment. The problem identification resulted in a novel design of an information system architecture, called TeamWork, for team-based task management in mobile business processes. TeamWork was implemented based on multi-agent technology and applied to a real mobile business process. The key contribution and novelty of this paper can be summarised as follows. Firstly, to the authors' knowledge, TeamWork is the first attempt to allow multiple TAMs to be interchangeably used for assigning tasks to workers. Secondly, the TTMM is the first model that abstracts the team dynamics including cooperative role and TAM management. Thirdly, this paper showed how TeamWork can be implemented using a distributed computing platform (a multi-agent platform).

There are further research issues as well. Work is underway to enhance the formality of TTMM so that it can be possible to check whether the dynamism of team management does not violate predefined organizational rules. A tool that allows developers to define TAMs using graphical editor is also being implemented.

References

Bauer, T. and Dadam, P. (2000). 'Efficient Distributed Workflow Management Based on Variable Server Assignments'. *Lecture Notes in Computer Science*, 1789: 94-109.

Bellifemine, F. et al. (2003). 'JADE—A White Article'. *Telecom Italia Lab J. Exp*, 3(3):6–19.

Lesaint, D., Voudouris, C. and Azarmi, N. (2000). 'Dynamic Workforce Scheduling for British Telecommunications plc'. *Interfaces*, 30(1): 45-56.

Davis R. and Smith R.G. 1983, 'Negotiation as a metaphor for distributed problem solving'. *Artificial Intelligence*, 20: 63-109.

Geppert, A. Kradolfer M. and Tombros, D. (1998). 'Market-Based Workflow Management'. *Lecture Notes in Computer Science*, 1402: 179-191.

Graves, S. C. 1981, 'A Review of Production Planning', *Operations Research*, 29(4): 647-675.

Harchol-Balter, M. (2002). 'Task Assignment with Unknown Duration'. *Journal of the ACM*, 49(2): 260-288.

Harchol-Balter, M., Crovella, M.E. and Murta, C.D.(1998). 'On Choosing a Task Assignment Policy for a Distributed Server System'. *Lecture Notes in Computer Science*, 1469: 231-242.

Lee, H., Mihailescu, P. and Shepherdson, J. 2003. 'Conversational Component-based Open Multi-agent Architecture for Flexible Information Trade'. *Lecture Notes in Artificial Intelligence*, 2782: 109-116.

Sandholm, T. 1995. 'An implementation of the Contract Net Protocol based on marginal cost calculations'. *Proceedings of the Eleventh National Conference on Artificial Intelligence*, Washington, D.C., AAAI/The MIT Press, Menlo Park, CA.

Shen M., Tzeng, G.H. and Liu, D.R. (2003). Multi-Criteria Task Assignment in Workflow Management Systems'. In *Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS'03)*.

Stonebraker, M., P.M. Aoki, R. Devine, W. Litwin and M. Olson, Mariposa: A New Architecture for Distributed Database System, *Working Paper*, available at http://mariposa.cs.berkeley.edu/.

Strobel M. 2000, 'On Auction as the negotiation paradigm of electronic markets: Success factors, Limitations and research directions,' *Electronic Markets*, 10(1): 39-44.

Tan, J.C. and Harker, P.T. (1999). 'Designing Workflow Coordination: Centralized Versus Market-Based Mechanisms', *Information Systems Research*, 10 (4): 328-342.

Varian, H.R., Economic Mechanism Design for Computerized Agents, *Working Paper, Department of Economics, University of Michigan*.