International Journal of Parallel, Emergent and Distributed Systems

Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/gpaa20

Method of using BDI agents to implement service-oriented workflow mapping in AGWMS

Lei Cao a, Jian Cao a, Minglu Li a & Feilong Tang a

a Shanghai Jiao Tong University, Department of Computer Science and Engineering, Shanghai, 200240, People's Republic of China
Published online: 04 Mar 2008.

To cite this article: Lei Cao, Jian Cao, Minglu Li & Feilong Tang (2008): Method of using BDI agents to implement service-oriented workflow mapping in AGWMS, International Journal of Parallel, Emergent and Distributed Systems, 23:2, 171-195

To link to this article: http://dx.doi.org/10.1080/17445760701659290

PLEASE SCROLL DOWN FOR ARTICLE

Full terms and conditions of use: http://www.tandfonline.com/page/terms-and-conditions

This article may be used for research, teaching, and private study purposes. Any substantial or systematic reproduction, redistribution, reselling, loan, sub-licensing, systematic supply, or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The accuracy of any instructions, formulae, and drug doses should be independently verified with primary sources. The publisher shall not be liable for any loss, actions, claims, proceedings, demand, or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.
Method of using BDI agents to implement service-oriented workflow mapping in AGWMS

Lei Cao*, Jian Cao1, Minglu Li2 and Feilong Tang3

Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, People’s Republic of China

(Received 20 March 2007; final version received 2 July 2007)

Workflow management is one of the most important parts of ShanghaiGrid, which has a complex service-oriented infrastructure. Because grid environment is very dynamic and services in it are shared among many users, it is impossible to optimise grid workflow from the point of view of execution ahead of time. In fact, one may want to make decisions about the execution locations and the access to a particular data set as late as possible, which is the main objective of grid workflow mapping. In this paper, three types of workflow are denoted as abstract workflow (AW), concrete workflow (CW) and executable workflow (EW). We propose a method of using Belief-desire-intention (BDI) agents to implement service-oriented workflow mapping in the agent-based grid workflow management system (AGWMS) of ShanghaiGrid. BDI agent technology helps the system meet challenges from the grid context. Contract net protocol (CNP) provides a very proper negotiation model for these agents. The problem of workflow mapping has been transferred to the problem of multi-agent negotiation with the help of CNP model in AGWMS. AW2CW and CW2EW mapping algorithms are also given to accomplish service-oriented workflow mapping.

Keywords: ShanghaiGrid; service-oriented; BDI agent; workflow mapping; contract net protocol

AMS Subject Classification: 68M14; 68T20

1. Introduction

With the advent of grid technologies [11], scientists, engineers and businessmen are building more and more complex applications to manage and process large data sets, and to execute respective tasks on distributed grid resources. Means of composing and executing distributed applications and services to form a complex workflow are needed. Some efforts in the research of business workflow and web services composition [4] can be reused in grid workflow system. Grid workflow is the workflow intended to solve sophisticated problems that occur in highly heterogeneous, distributed, complex and dynamic grid environments that comprise of one or more virtual organisations (VOs). Given the dynamic nature of grid environments, it is more suitable for users to define workflow applications with the abstract model, in which the workflow is specified without referring to specific grid resources for task execution. Obviously, the mapping task from abstract workflow (AW) to concrete workflow (CW) [7,9], which has been bound
to specific grid resources, might be completed by a grid workflow management system. It’s a complicated matchmaking process. Because users might define various tasks with unfixed styles in AW model according to various demands, we could not do matchmaking only by comparing between task demands and service functionalities. As the autonomic entity, software agent can do many complicated tasks by following its inner plans. These agents can also negotiate with each other to accomplish more complicated tasks. Merging of service with agent is a technical trend in computer areas. Many agents will represent corresponding tasks in the service-oriented workflow and cooperate with each other to achieve the goal of the workflow. The job to find proper agents to match tasks might be done by negotiations among agents. Contract network protocol (CNP) [3,19] is one kind of classical negotiation protocol in distributed artificial intelligence (DAI) area. CNP is also one of the most popular controlling structures used in multi-agent system (MAS), and it has been applied to some actual systems, e.g. Agile manufacturing system, air traffic management system [1,2].

In this paper, we propose a method of using Belief-desire-intention (BDI) agents to implement service-oriented workflow mapping, and apply it to the agent-based grid workflow management system (AGWMS) of ShanghaiGrid. Three types of workflow in AGWMS are denoted as AW, CW and executable workflow (EW). Utilisation of them further liberates model designers from sophisticated system details. It also increases the reusability of domain-specific models and improves the fault tolerance of the system. BDI agent technology is used to meet challenges from the grid context. The mapping from AW to CW is actually the process of multi-agent negotiation to get planning agent (PA) union as the concrete substitute of AW. CNP is adopted as the explicit interaction model for agent negotiation, which is one of the emphases of this paper. We also propose AW2CW and CW2EW mapping algorithms to accomplish service-oriented workflow mapping.

The remainder of the paper is organised as follows. Section 2 describes the related work. Section 3 gives an overview of AGWMS of ShanghaiGrid. Section 4 introduces a mapping case. Section 5 gives more details about BDI agent method in the process of service-oriented workflow mapping. Section 6 presents the implementation of the system. We conclude in Section 7 with lessons learned and future research considerations.

2. Related work

Many efforts toward mapping workflow tasks onto specific grid resources have been made. In Ref. [21], authors propose a workflow enactment engine (WFEE) with a just in-time scheduling system using tuple spaces. It allows the decision of resources allocation to be made dynamically at the time of the execution of tasks in the workflow, so tasks are only concreted at execution time. Only then, the CW model could be gotten and its tasks will be bound to specific resources. This measure adapts to the dynamic grid environment. But considerations about quality of service (QoS) are lacked in the procedure of getting CW model.

The mapping of Grid workflows onto Grid resources based on existing planning technology is presented in Ref. [8]. This work focuses on coding the problem to be compatible with the input format of specific planning systems and thus transferring the mapping problem to the planning problem. Although this is a flexible way to gain different destinations, significant disadvantages regarding the time-intensive computation, long response time and missing consideration of Grid-specific constraints
appeared. The latter is the main cause that the suggested solutions often do not express the expected quality.

Cao et al. presented an algorithm that maps each sub job separately on individual Grid resources in Ref. [5]. The algorithm processes one sub job at one time, schedules it to a suitable RMS with start time slot not conflicting the dependency of the flow. The selection of the destination resources is optimised with respect to a minimal completion time. When applying this strategy to the specified problem, each sub job will be assigned separately to the cheapest feasible resource management system (RMS). This strategy allows fast computation of a feasible schedule, but it lacks considerations of dependencies between sub jobs of the workflow from the global viewpoint.

CNP is adopted as the explicit interaction model for agent negotiation. Smith [19] and Davis [20] proposed the classical CNP at the beginning of 1980s last century. It utilises market-similar mechanism of BidInviting-bidding-contracting. By the manner of task publication, it solves the allocation problem about distributed dynamic tasks with negotiations and competitions among nodes in the net. But the classical CNP has some weaknesses. It gives no consideration on collision identifying and solving because it omits existing conflicts between agents. In the phase of bid inviting, all nodes could take part in the process. The manager has to evaluate large numbers of tenders from bidders, thus the communication traffic is very heavy and more resources are needed, which will cause the problems of bottleneck and information congestion on the manager side. Aiming at such weaknesses, we utilise an improved CNP model in the process of grid workflow mapping to get better efficiency and higher success rate.

3. Overview of AGWMS

ShanghaiGrid [15] is the first metropolitan grid in China. The project aims to develop a platform to share distributed resources in Shanghai conveniently based on the grid technology. ShanghaiGrid has a complex service-oriented infrastructure. As one part of ShanghaiGrid, we have developed an AGWMS, which helps grid users streamline, manage and monitor their routinely problem solving processes without having to know any details of the underlying complex structure and dynamic state of VOs. Three types of workflow in AGWMS are denoted as AW, CW and EW. AW describes the whole functionalities of the process, so it focuses on ‘What to do’. In AGWMS, AW is composed of many PAs and it does not refer to any actual services. CW presents the detailed implementation steps of the process, so it focuses on ‘How to do’. In AGWMS, CW is composed of logical nodes, internal actions and service agents (SAg). Each SAg corresponds to a list of services, which have the same functionality, but it does not bind to any of them. EW selects the final executors in the process, so it focuses on ‘Who to do’. In AGWMS, EW can be gotten after all SAgS in CW bind to proper services.

3.1 Architecture of AGWMS

From the Figure 1, you can see that there are four layers in AGWMS. The user layer is the interface between user and system. There are mainly two modules in this layer: workflow modelling tool and workflow console. The engine layer is the core layer to manage workflow execution. The workflow engine is distributed using JINI technology [14]. It’s based on event-condition-action (ECA) [12] rules. The agent layer makes it possible that the workflow could be divided into three types: abstract, concrete and executable. The adapter layer makes it easy that AGWMS invokes external applications. Adapter is the
proxy for the invokable application. It encapsulates the complex invoking details. Different types of adapter are designed in AGWMS.

### 3.2 BDI agents in AGWMS

All agents in the agent layer are constructed according to BDI model [17]. There are six kinds of agents in the layer depicted in Figure 2. The model is based on Jadex [13]. The term ‘plan’ in Jadex is modified to ‘project’ in this paper to be distinguished from our own ‘plan’ notation. The term ‘capability’ is used for different purposes in the agent community. In the context of AGWMS, the term is used to denote an encapsulated agent module composed of beliefs, goals and projects. A capability is basically the same as an agent, but without its own reasoning process. Plan is a special kind of workflow, which is used in PA of AGWMS. Plan is the internal expression of an external capability of PA. Each plan has many common components compared with the generic workflow, such as logical nodes, internal actions (*start, finish, delay, evaluate, convert*). But external actions,
such as invocations of Web or Grid Services, operations on database or execution of other kinds of application, will be replaced with application agents (AAs) in the plan. In service-oriented context, AA refers to SAg. Beliefs represent the agent’s knowledge about its environment and itself. In AGWMS, the beliefs can be any Java objects. They are stored in a belief base and can be referenced in expressions, as well as accessed and modified from projects using the belief base interface. Goals make up the agent’s motivational stance and are the driving forces for its actions. Therefore, the representation and handling of goals is one of the main features of AGWMS. The goal notation is equal to the desire notation. Projects represent the agent’s means to act in its environment. Therefore, the projects predefined by the developer compose the library of (more or less complex) actions the agent can perform. Depending on the current situation, projects are selected in response to occurring events or goals. The project notation is equal to the intention notation. Table 1 presents the concrete definitions about the FIPA-compatible [10] BDI agent model of AGWMS.

There are six kinds of agent in the BDI agent model of AGWMS (Figure 2).

3.2.1 Planning agent

PA is the main component of AW. It is domain specific, and its capabilities are always stable and predefined. In some sense, PA takes the role of host of sub-workflow denoted in business workflow domain. In this paper, each capability of PA represents a plan. Each PA has multiple plans in its plan library, so it has multiple capabilities in its capability library. PA is also responsible for the scheduling of plan execution. It contains the agent engine to carry out the scheduling.

$$\text{Plans}(PA) = \{plan_1, plan_2, \ldots, plan_n\}$$

$$\text{Capability}(PA) = \{\text{Capability}(plan_1), \text{Capability}(plan_2), \ldots, \text{Capability}(plan_n)\}$$

$$\text{Beliefs}(\text{Capability}(plan_i)) = \{\text{Preconditions}(plan_i), \text{Effects}(plan_i), \text{FunctionalityName}(plan_i)\}.$$
3.2.2 Service agent

SAg is the main component of PA plan. It corresponds with a list of service adapters that have the specific function. A service adapter might bind some services having same functionality, whereas services from different service adapters of the same SAg are only similar in functionality. Each capability in SAg represents one service adapter. SAg has multiple service adapters in its adapter library, so it has multiple capabilities in its capability library. In the mapping process of CW to EW, SAg is responsible for binding

### Table 1. BDI definitions of agents in SHGWMS.

<table>
<thead>
<tr>
<th>Agent</th>
<th>Beliefs</th>
<th>Desires</th>
<th>Intentions</th>
</tr>
</thead>
<tbody>
<tr>
<td>UA</td>
<td>AdA address</td>
<td>Goal to perform mapping of AW</td>
<td>Project of mapping AW</td>
</tr>
<tr>
<td></td>
<td>MA address</td>
<td>Goal to perform CW execution</td>
<td>Project of executing CW</td>
</tr>
<tr>
<td>AdA</td>
<td>IA address</td>
<td>Goal to perform mapping of CW</td>
<td>Project of mapping CW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Goal to perform EW execution</td>
<td>Project of executing EW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Goal to query CW information</td>
<td>Project of query</td>
</tr>
<tr>
<td>MA</td>
<td>IA address</td>
<td>Goal to perform bids invitation to PAs</td>
<td>Project of bids invitation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Goal to achieve CW composition</td>
<td>Project of CW composition</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Goal to query PAs information</td>
<td>Project of query</td>
</tr>
<tr>
<td>IA</td>
<td>Important data of all agents</td>
<td>Goal to maintain all agents information</td>
<td>Project of maintaining all agents information</td>
</tr>
<tr>
<td></td>
<td>Database access point</td>
<td>Goal to return query result</td>
<td>Project of returning query result</td>
</tr>
<tr>
<td>PA</td>
<td>Preconditions of plan</td>
<td>Goal to execute plan</td>
<td>Project of executing plan with the agent engine</td>
</tr>
<tr>
<td></td>
<td>Effects of plan</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Functionality name of plan</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PA</td>
<td>IA address</td>
<td>Goal to perform bids submission</td>
<td>Project of bids submission</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Goal to perform capability searching</td>
<td>Project of searching demanded capability</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Goal to perform new capability generation</td>
<td>Project of generating new capability</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Goal to perform capability updating in IA</td>
<td>Project of updating capability in IA</td>
</tr>
<tr>
<td>SAg</td>
<td>Preconditions of service adapter</td>
<td>Goal to execute service adapter</td>
<td>Project of executing service adapter</td>
</tr>
<tr>
<td></td>
<td>Effects of service adapter</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Functionality name of service adapter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAg</td>
<td>IA address</td>
<td>Goal to perform capability updating in IA</td>
<td>Project of updating capability in IA</td>
</tr>
</tbody>
</table>


### Notes

1. Preconditions of plan
2. Effects of plan
3. Functionality name of plan
the most suitable service and uniting others to make the whole workflow have the maximum QoS value.

\[
\text{Adapters}(S\text{Ag}) = \{\text{adapter}_1, \text{adapter}_2, \ldots, \text{adapter}_n\}
\]

\[
\text{Capability}(S\text{Ag}) = \{\text{Capability}(\text{adapter}_1), \text{Capability}(\text{adapter}_2), \ldots, \\
\text{Capability}(\text{adapter}_n)\}
\]

\[
\text{Beliefs}(\text{Capability}(\text{adapter}_i)) = \{\text{Preconditions}(\text{adapter}_i), \text{Effects}(\text{adapter}_i), \\
\text{FunctionalityName}(\text{adapter}_i)\}.
\]

3.2.3 Information agent

Information agent (IA) is the registry of all agents in the multi-agent platform. We adopt multi-layer distributed framework for IA in Grid environment. Each VO has its local IA. The local IA keeps the information about capabilities of all PAs and SAGs in the VO. SAGs are chosen preferentially from the same VO when building a plan in order to get better efficiency at the execution time. The general IA has the registration information about all local IAs.

3.2.4 Merging agent

Merging agent (MA) is responsible for merging all related plans of PAs to get a CW according to the predefined AW. Based on all functionality modules (FM) defined in AW, MA will send out bid invitations to related PAs. Both Plan generating algorithm and Plan merging algorithm are utilised by MA.

3.2.5 User agent

As the representative of users, User agent (UA) interacts with workflow console and Macro-workflow modelling tool. UA gets AW model from Macro-workflow modelling tool and returns corresponding CW latterly. UA also receives CW execution command from workflow console and returns real-time execution states latterly. UA would interact with other agents in the system to complete these tasks.

3.2.6 Admin agent

Admin agent (AdA) is responsible for the mapping from CW to EW using Genetic algorithm (GA). Furthermore, it is responsible for the scheduling of EW execution. It contains the global engine to carry out the scheduling.

4. A mapping case

In order to show how BDI agents are used to implement service-oriented workflow mapping, this section introduces a service-based system model for Online conference arrangement, which needs to integrate the services provided by different organisations in ShanghaiGrid environment such as Conference organisers, Ticket agents, Car rental companies, Hotels, Restaurants, Travel agencies and Retailers. We need build the
domain-specific ontology for workflow mapping in advance. Plan generation and selection, PA selection, SAg selection and service adapter selection are all based on the term ontology.

4.1 Abstract workflow model
We assume that several international conferences will be held at Shanghai Jiao Tong University this year and the conference organisers want to build an online conference arrangement service for the future attendee. Our AGWMS provides them the platform to build such composite service, which is actually a service-oriented workflow. At first step, they build an AW model using Macro-workflow modelling tool (Figure 3). Based on their experiences and the domain ontology given, the conference organisers define the properties and requirements of each FM in the model, e.g. name, functionality, maximum permitted execution price, execution time and maximum permitted solution quality (Table 2). They also prescribe the preconditions and effects of these FMs. The expressions within the brackets of Functionality column in Table 2 are names of corresponding atomic actions of FM functionality. Thus, there is no more work that the user should be concerned with. The mapping job of AW to CW and CW to EW will be leaved to the system.

4.2 Present BDI agents
For the reason of clearness, we give BDI details only about present PAs of the domain in this case. Other types of agent, such as MA, IA, UA and SAGs are not referred to here. Fifteen PAs are assumed in the system. They are representatives of fifteen companies and able to implement complex tasks autonomously. Each of them has one plan at least. Each plan is corresponding to one capability in PA. Table 3 describes them. We take Registration-A0 PA as the example. It has two plans in its plan library: Conf_reg and Visit_reg (Figure 4). The corresponding capability ConfRegistration is given in the file ConfRegistration.capability.xml and the other capability VisitRegistration is given in the file VisitRegistration.capability.xml. Registration-A0 PA itself is configured in the file RegistrationPA.agent.xml.

4.3 Key problems
Till now, we possess both the requirements of all FMs in the AW model and the capabilities of all PAs in the domain. They should have the same representation basis,
<table>
<thead>
<tr>
<th>FM name</th>
<th>Functionality</th>
<th>Constrains: price/time/quality (RMB/s/%)</th>
<th>Preconditions</th>
<th>Effects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Register_Conference</td>
<td>Registration of conference {Conf_Inquiry, Attendee_Registration, Documents_Distribution}</td>
<td>20/60/80</td>
<td>Conf_name, Attendee_name</td>
<td>Conf_starttime, Conf_endtime, Conf_city, Registration_id</td>
</tr>
<tr>
<td>Book_Ticket</td>
<td>Booking of ticket {TT_Inquiry, TT_Select, TT_Booking}</td>
<td>50/80/80</td>
<td>Source_city, Destination_city, Start_date, Passenger_name, Vehicle_no, Vehicle_type, Vehicle_no, Ticket_cost, Start_time, Arrival_time, Booking_id, Rental_id</td>
<td>Rental_cost, Rental_ID</td>
</tr>
<tr>
<td>Rent_Car</td>
<td>Rental of car {CR_Inquiry, CR_Select, CR_Rent}</td>
<td>500/600/80</td>
<td>Rental_city, Renter_name, Drivelicense_ID, Car_type, Rental_starttime, Rental_endtime, Rental_endlocation</td>
<td>Rental_cost, Rental_ID</td>
</tr>
<tr>
<td>Reserve_Accommodation</td>
<td>Reservation of Accommodation {Hotel_Search, Hotel_Select, Hotel_ Reserve, Restaurant_Search, Restaurant_Select, Restaurant_ Reserve}</td>
<td>600/400/80</td>
<td>Conf_city, Attendee_name, Hotel_level, Restaurant_type, Enter_date, Leave_date, Source_city, Destination_city, Traveler_name</td>
<td>Hotel_name, Hotel_address, Restaurant_name, Restaurant_address, Reservation_id, Duration, Total_cost, Tour_Details, Tour_cost, Arrangement_ID</td>
</tr>
<tr>
<td>Arrange_Tour</td>
<td>Arrangement of tour {LT_Search, LT_Select, LT_Register}</td>
<td>300/1800/80</td>
<td>Source_city, Destination_city, Traveler_name</td>
<td>Total_cost, Tour_Details, Tour_cost, Arrangement_ID</td>
</tr>
<tr>
<td>Buy_Souvenir</td>
<td>Buying of souvenir {Goods_Query, Goods_Purchase, Goods_Deliver}</td>
<td>500/3000/80</td>
<td>Preferring_type, Customer_name, Delivery_address</td>
<td>Total_cost, Order_ID</td>
</tr>
<tr>
<td>PA name</td>
<td>Capabilities</td>
<td>Plans</td>
<td>Note</td>
<td>Cost/time (RMB/s)</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------------</td>
<td>------------------</td>
<td>----------------------------------------------------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Registration-A0</td>
<td>ConfRegistration</td>
<td>Conf_reg</td>
<td>Registration of conference by company A0</td>
<td>15/45</td>
</tr>
<tr>
<td></td>
<td>VisitRegistration</td>
<td>Visit_reg</td>
<td>Registration of visit by company A0</td>
<td>20/40</td>
</tr>
<tr>
<td>Registration-B0</td>
<td>ConfRegistration</td>
<td>Conf_reg</td>
<td>Registration of conference by company B0</td>
<td>18/50</td>
</tr>
<tr>
<td></td>
<td>VisitRegistration</td>
<td>Visit_reg</td>
<td>Registration of visit by company C0</td>
<td>22/50</td>
</tr>
<tr>
<td>Ticket_Booking-A1</td>
<td>TranspBooking</td>
<td>TranspTicket_booking</td>
<td>Transportation ticket booking by ticket-agency A1</td>
<td>40/70</td>
</tr>
<tr>
<td></td>
<td>EntertBooking</td>
<td>EntertTicket_booking</td>
<td>Entertainment ticket booking by ticket-agency A1</td>
<td>10/30</td>
</tr>
<tr>
<td></td>
<td>MatchBooking</td>
<td>MatchTicket_booking</td>
<td>Match ticket booking by ticket-agency A1</td>
<td>10/30</td>
</tr>
<tr>
<td>Ticket_Booking-B1</td>
<td>EntertBooking</td>
<td>EntertTicket_booking</td>
<td>Entertainment ticket booking by ticket-agency B1</td>
<td>30/60</td>
</tr>
<tr>
<td></td>
<td>MatchBooking</td>
<td>MatchTicket_booking</td>
<td>Match ticket booking by ticket-agency B1</td>
<td>10/20</td>
</tr>
<tr>
<td>Register&amp;Booking_SP-C1</td>
<td>ConfRegistration &amp;TranspBooking</td>
<td>Ticket_booking</td>
<td>Registration of conference and booking transportation tickets by Register&amp;Booking_SP-C1.</td>
<td>50/80</td>
</tr>
<tr>
<td>Rental_SP-A2</td>
<td>CarRental</td>
<td>Car_rental</td>
<td>Rental of car in renter A2</td>
<td>450/500</td>
</tr>
<tr>
<td></td>
<td>BicycleRental</td>
<td>Bicycle_rental</td>
<td>Rental of bicycle in renter A2</td>
<td>50/180</td>
</tr>
<tr>
<td></td>
<td>CopterRental</td>
<td>Copter_rental</td>
<td>Rental of copter in renter A2</td>
<td>20,000 /3600</td>
</tr>
<tr>
<td></td>
<td>YachtRental</td>
<td>Yacht_rental</td>
<td>Rental of yacht in renter A2</td>
<td>15,000 /3600</td>
</tr>
<tr>
<td>Rental_SP-B2</td>
<td>CarRental</td>
<td>Car_rental</td>
<td>Rental of car in renter B2</td>
<td>480/540</td>
</tr>
<tr>
<td></td>
<td>BicycleRental</td>
<td>Bicycle_rental</td>
<td>Rental of bicycle in renter B2</td>
<td>45/300</td>
</tr>
<tr>
<td>Hotel_Reservation-A3</td>
<td>HotelReservation</td>
<td>Hotel_reservation</td>
<td>Reservation of hotel A3</td>
<td>200/180</td>
</tr>
<tr>
<td>Hotel_Reservation-B3</td>
<td>HotelReservation</td>
<td>Hotel_reservation</td>
<td>Reservation of hotel B3</td>
<td>220/200</td>
</tr>
<tr>
<td>RestaurantReservation-A4</td>
<td>RestaurantReservation</td>
<td>Restaurant_reservation</td>
<td>Reservation of restaurant A4</td>
<td>180/180</td>
</tr>
<tr>
<td>RestaurantReservation-B4</td>
<td>RestaurantReservation</td>
<td>Restaurant_reservation</td>
<td>Reservation of restaurant B4</td>
<td>200/240</td>
</tr>
<tr>
<td>RestaurantReservation-C4</td>
<td>RestaurantReservation</td>
<td>Restaurant_reservation</td>
<td>Reservation of restaurant C4</td>
<td>250/240</td>
</tr>
<tr>
<td>Tour_Arrangement-A5</td>
<td>OverseasTour</td>
<td>Overseas_tour</td>
<td>Overseas tour arrangement of tour-agency A5</td>
<td>15,000/2400</td>
</tr>
<tr>
<td></td>
<td>DomesticTour</td>
<td>Domestic_tour</td>
<td>Domestic tour arrangement of tour-agency A5</td>
<td>3000/1800</td>
</tr>
<tr>
<td></td>
<td>LocalTour</td>
<td>Local_tour</td>
<td>Local tour arrangement of Tour-agency A5</td>
<td>250/1200</td>
</tr>
<tr>
<td>Tour_Arrangement-B5</td>
<td>DomesticTour</td>
<td>Domestic_tour</td>
<td>Domestic tour arrangement of tour-agency B5</td>
<td>2500/2000</td>
</tr>
<tr>
<td></td>
<td>LocalTour</td>
<td>Local_tour</td>
<td>Local tour arrangement of Tour-agency B5</td>
<td>200/1500</td>
</tr>
</tbody>
</table>
on which the comparison between them could be achievable. Before the matchmaking between requirements and capabilities, how to represent them in the proper way is the first key problem.

As we can see in the definition of AW model, all its FMs are not only independent but also relative to each other. Relations among agents in MAS are coincident with this situation. How to do matchmaking between FM requirements and PA capabilities is the second key problem.

From multiple PA capabilities that can satisfy the requirements of FM, how to choose the most suitable candidate for the substitution is the third key problem.

The algorithms of using BDI agents depicted in the next section can solve all these key problems rightly.

5. Algorithms in the mapping process

5.1 Some entities’ formalisation

Domain ontology specifies meanings of all objects in the application domain and also gives their various relations, such as subsumption relation (written as Is-a), meronymy relation (written as part-of), reference relation (written as applied-to) and other types. Construction of AW and definitions of its FMs, whose properties include name, functionality, precondition, effect, price, Time and QoS, are all based on the ontology. Taking Conference_Arrangement AW as the example (Figure 3), we might give the formalisation of the AW model. It has six FMs. Each of them has been given respective requirements. Execution sequence and data flow are also defined. The formalisation contains two parts: a set of FMs and the constitution of AW. The later includes AW steps, a set of ordering constraints and a set of causal links. Details are omitted here.

5.2 CNP-based negotiation procedure

Our research on task negotiation is based on the multi-agent platform given in Section 2.2. Three types of agent: UA, MA and PA share the same responsibility of workflow mapping and thereby form an agent group. There are two layers of negotiation. In the first layer, UA consigns user’s mapping request to MA. We choose the simple direct-consigning strategy, by which UA directly consigns the mapping request to the corresponding MA in the task domain. In the second layer, MA cooperates with proper PAs to accomplish the decomposition of mapping task. We choose modified CNP model as the strategy. The negotiation procedure contains three processes: bid-inviting, bidding and contracting. Details are given below.
5.2.1 Bid-inviting process

After receiving the mapping request from Macro-workflow modelling tool, UA sends out the AW model to MA. MA will transform it into multiple FM sub-bids based on the relationships among FMs in the AW. Then it will put these sub-bids together as an AW bid. The bid’s specification is given in Figure 5.

‘Bid deadline’ item means the useful life of the AW bid. Dependency of sub-bids could be gotten from the set of ordering constraints of AW. In the front part of sub-bid, there are items from definitions of FM, e.g. name, note and functionality requirements. MA also adds some constrains in the sub-bid, e.g. expected bid price, expected finish time and expected solution quality. The first two are given from the view point of Manager – MA and the last one is given from the view point of Contractor – PA.

To determine the expected bid price $P_i$ is the key step. Some influencing factors include:

- Character preference of MA;
- Maximum FM price $H\_PRICE$ given by user;
- Constrains of task;
- Idle state of PA.

We give the definition: $P_i = H\_PRICE \cdot (1 - \alpha), 0 < \alpha < 1$. The symbol $\alpha$ is an undermined coefficient. Its value is affected by these factors given.

Expected finish time $T_i$ is actually the $TIME$ element in the FM’s definition. In addition, we quantise the solution quality by giving the definition $E_i$ in single FM substitution case:

$$E_i = \frac{\text{ActionNum}(FM)}{\text{ActionNum}(Plan)}$$

$\text{ActionNum}(FM)$ – Number of atomic actions’ name in the FM functionality;

$\text{ActionNum}(Plan)$ – Number of actions in PA plan.

Capability-efficiency ration $E_i$ denotes the utilisation degree of PA capability when the corresponding plan could substitute the FM. Obviously, $E_i \leq 1$ is always correct. The expected constrain of solution quality given in the sub-bid is actually the $QOS$ element in the FM’s definition.

![Figure 5. Specification of AW bidding document.](image)
As introduced in Section 2.2, IA preserves important information of all PAs in the VO, e.g. addresses, capabilities. MA could get all related information of proper PAs belonging to the application domain by querying. After the AW bid document is made, MA will broadcast it to these PAs.

5.2.2 Bidding process

After PA receives the AW bid, it will analyse all FM sub-bids and make corresponding tender considering its capabilities and other factors, e.g. current charged tasks, possible bidding benefits.

Decisive factors of PA bidding contain:

- Whether PA capability could satisfy functionality requirements given in the FM sub-bid;
- Current charged tasks of PA;
- Whether PA could satisfy constrains of expected solution quality and expected execution time;
- We define the actual cost of FM substitution by PA plan as \( \text{Cost}(\text{Plan}) \). If the inequality \( \text{Cost}(\text{Plan}) > K \times P_t \) is true (\( K \) is the beneficial coefficient), PA will not bid it.

Having the intention to bid the FM, PA will make the tender for it. The tender price \( P_p \) is an important element. It is calculated as: \( P_p = \text{Cost}(\text{Plan}) \times (1 + \beta) \), \( 0 < \beta < 1 \). \( \beta \) is the undermined coefficient. Its value is affected by these factors given above.

The remainder of the section will focus on the problem on satisfaction of functionality requirements.

If we want to know whether one capability could match the functionality requirements of a FM, we could follow such steps:

- Firstly, make sure that the FM’s requirements about \( \text{Precondition} \) and \( \text{Effect} \) are satisfied by the corresponding plan’s \( \text{Precondition} \) and \( \text{Effect} \). That is:

\[
\text{PRECOND}(\text{FM}) \supseteq \text{PRECOND}(\text{Plan})
\]

\[
\text{EFFECT}(\text{FM}) \subseteq \text{EFFECT}(\text{Plan})
\]

- If satisfied, one property of the FM: Functionality(\( \text{FM} \)) will be decomposed into some proper names of atomic actions based on the domain ontology. According to these names, we should look for relative actions in action library of the plan. If all names could have their corresponding actions, we could say that the PA capability could satisfy functionality requirements of the FM.

There are mainly four kinds of matchmaking results between PA capability and functionality requirements:

1. \( \text{Just-satisfied} \): Corresponding plan of PA capability has the just functionality to substitute the FM. In this situation, \( E_t = 1 \). PA will bid it and suggest the substitution by the plan;
2. \( \text{Partly-satisfied} \): PA plan only has partial functionality of the FM. In this situation, no need to calculate \( E_t \). PA will bid it and explain the satisfied part in the tender;
3. \( \text{Not-satisfied} \): PA plan has none of the functionality of the FM. PA will not bid it;
(4) **Over-satisfied:** Merely one part of PA plan could have the equal functionality of the FM. In this situation, $E_f < 1$. If multiple FMs’ functionalities are all over-satisfied by the plan, PA will follow the *Three Principles of Multi-FM substitution* and deal with them. Otherwise, PA will calculate $E_f$ and compare it with the constrain value of solution quality given in the FM sub-bid. If the calculated value is less than the constrain value, PA will not bid it. Or else, PA will bid it and suggest the substitution by the plan.

In order to decrease communication traffic among agents in the execution phase, and to increase manageability of optimisation about multi-FM functionalities by one agent, we make the *Three principles of Multi-FM substitution* as following (Figure 6).

1. **Continuous principle:** Multiple FMs, which are satisfied by the same plan, must be continuous in the AW.
2. **No-branch principle:** In multiple FMs, which are satisfied by the same plan, no branch in the middle is allowed to link with other FMs.
3. **Symmetrical principle:** If multiple FMs, which are satisfied by the same plan, exist in different execution paths (parallel or condition type), they must be symmetrical. That is to say, FMs in different paths must symmetrically contain the start point of branches or not. Moreover, they must symmetrically contain the end point of branches or not.

After PA deals with all FM sub-bids based on the *Three Principles*, these eligible FMs that are continuous and satisfied by the same plan will form into a temporary $AW_0$. We give the definition of capability-efficiency ration $E_a$ in the multi-FM substitution:

$$E_a = \frac{\text{ActionNum}(AW_0)}{\text{ActionNum}(\text{Plan})}$$

- $\text{ActionNum}(AW_0)$ — Number of atomic actions’ name in the set of $\text{Functionality}(AW_0)$;
- $\text{ActionNum}(\text{Plan})$ — Number of actions in PA plan, which will be the substitution of multi-FM.

Capability-efficiency ration $E_a$ denotes the utilisation degree of PA capability when the corresponding plan could substitute multiple FMs. It is obviously that the inequation $E_a \leq 1$ is correct. If $E_a$ is bigger than the minimum value in all FMs’ solution quality constrains, PA will choose the plan to substitute these FMs and bid them with a single tender. It will also attach the number of actions of the set of $\text{Functionality}(AW_0)$ as the

![Figure 6. Principles of multi-FM substitution.](image-url)
choosing criterion of the best substitution plan used by MA. Otherwise, PA will give up
the bidding intention to these FM sub-bids.

After all FM sub-bids are dealt with as above situations, PA will make sub-tenders and
gather them to form a single tender. Figure 7 shows the specification of the tender.

The final tender sent by PA to MA shows the bidding results aiming at all FMs in the
AW. According to the analysis given above, three types of sub-tender are given:

1. Sub-tender aiming at single FM whole-substitution (type = 0);
2. Sub-tender aiming at single FM part-substitution (type = 1);
3. Sub-tender aiming at multi-FM substitution (type = 2).

Their concrete structures can be seen from the Figure 7. Related explanations about
them have been given in above paragraphs.

5.2.3 Contracting process

When the deadline is due, MA starts to evaluate all tenders for the best group of PA plans
as the final substitution of the AW. There are many kinds of factors influencing the
selection, e.g. execution cost, solution quality, PA communication traffic. Combining with
these factors, we define two optimisation rules for PAs, which have satisfied requirements
of FM sub-bids.

1. Maximum FM substitution rule: MA will choose the PA capability that can
substitute the most of FMs. Thus PA number in the final CW could be reduced, and
corresponding PA communication traffic will decrease accordingly at the
workflow execution time.

2. Maximum MA benefit rule: based on the Maximum FM substitution rule, MA will
select the best one from the eligible PA capabilities to get the most benefits, e.g.
lowest price, shortest execution time and best solution quality.

---

Figure 7. Specification of bid proposals.
We define the compositive expression of Benefits as below:

$$\text{Benefits}_{MA} = w_1 \times \left[ \frac{(P_f - \text{Cost}_{PA})}{P_f} \right] + w_2 \times \left[ \frac{(T_f - \text{Time}_{PA})}{T_f} \right] + w_3 \times \left[ \frac{(QoS_{PA} - E_t)}{QoS_{PA}} \right]$$

$$w_1 + w_2 + w_3 = 1.$$ 

The symbols, $P_f, T_f, E_t$, have been given definitions in the former sections. $\text{Cost}_{PA}$ stands for Bidding price of the PA tender, and it actually equals to $P_p$. $\text{Time}_{PA}$ stands for Finish time of the PA tender. $\text{QoS}_{PA}$ stands for Solution quality of the PA tender. Coefficients $w_1, w_2, w_3$ represent the element weights of Cost, Time and Solution quality in MA benefits respectively. Their values are determined by MA depending on its preferences. The default values are given: $w_1 = 0.5, w_2 = 0.3, w_3 = 0.2$.

From Section 4.2.2, we know that there are three types of tender from PA. Corresponding MA disposals are given as below:

1. The sub-tender aiming at multi-FM substitution has the priority to be dealt with. Choice of the PAs, which satisfy requirements of multiple FM sub-bids and correlate with each other in functionality, might be grouped into three situations as below (Figure 8):
   (a) When multiple PAs overlap in functionality, MA will choose the PA that bids the most of FMs (Maximum FM substitution rule). If they have the same number of FM, MA will select the most beneficial PA for itself (Maximum MA benefit rule);
   (b) When there are comprising relationships among multiple PAs in functionality, MA will choose the PA that bids the most of FMs (Maximum FM substitution rule);
   (c) When there are repetition relationships among multiple PAs in functionality, MA will choose the most beneficial PA for itself (Maximum MA benefit rule).

![Figure 8. Classification of multiple satisfied FM.](image_url)
When MA chooses the best PA as the bid-winner, it will send the contract of multi-FM substitution to the PA.

2. After disposing of all sub-tenders aiming at multi-FM substitution, MA will deal with sub-tenders aiming at single FM whole-substitution. MA considers their execution time, bidding price and solution quality in one context. After MA chooses the best PA as the bid-winner, it will send the single FM substitution contract to the PA.

3. Certainly, there might be a possibility that some FMs in AW do not have corresponding whole-substitution tenders. In this situation, MA has to analyse each FM’s part-substitution tenders received and judge whether there exists a PA bidding-union being able to satisfy all requirements of the FM. One of the criterions about eligible PA bidding-union is given as below:

\[
\begin{align*}
(PRECOND(Cap_1) \cup PRE\pmCOND(Cap_2) \cup \cdots \cup PRECOND(Cap_m)) \\
\subseteq PRECOND(FM)
\end{align*}
\]

\[
\begin{align*}
(EFFECT(Cap_1) \cup EFFECT(Cap_2) \cup \cdots \cup EFFECT(Cap_m)) &\supseteq EFFECT(FM)
\end{align*}
\]

\[
\begin{align*}
(Actions(Cap_1) \cup Actions(Cap_2) \cup \cdots \cup Actions(Cap_m)) &\supseteq Functionality(FM)
\end{align*}
\]

\(Actions(Cap_i)\) stands for the name set of actions in the selected capability \(Cap_i\) of the PA. The bidding price of the union is the summation of all members’ costs, and the bidding execution time is also the summation of all members’ execution time. The capability-efficiency ration of the PA bidding-union is also calculated to judge its eligibility: \(E_u = \frac{\text{ActionNum}(FM)}{\text{ActionNum}(PA\text{Union})}\).

\[
\begin{align*}
\text{ActionNum}(FM) &\quad \text{Number of the atomic actions in FM functionality;}
\end{align*}
\]

\[
\begin{align*}
\text{ActionNum}(PA\text{Union}) &\quad \text{Number of the atomic actions in the PA bidding-union}
\end{align*}
\]

Capability-efficiency ration \(E_u\) denotes the utilisation degree of the PA bidding-union’s capability when the PA bidding-union could substitute the FM. It is obviously that the inequation \(E_u \leq 1\) is correct. We assume that the solution quality of the FM has a constrain value \(e'\). When the inequation \(E_u \geq e'\) is correct, MA considers that the PA bidding-union is the eligible one to substitute the FM. Then MA sends the request of signing joint-bidding contract to each member of the PA bidding-union. If all PAs agree with the request, the FM will be substituted by the PA bidding-union; when the inequation \(E_u < e'\) is correct, MA considers the PA bidding-union is not eligible to substitute the FM. MA will abort the single FM substitution attempt.

Until now, MA has cooperated with proper PAs to accomplish the most of mapping job. But there might be some FMs having no corresponding PA plan to substitute. In addition, for these PAs that have won the bids, how to combine their corresponding plans together to form the final CW is also a critical problem. Facing with these situations, AW2CW mapping algorithm is proposed in the next section.

### 5.3 Mapping algorithms

The mapping from AW to CW is actually the process of multi-agent negotiation to get PA union as the concrete substitute of AW. Figure 9 gives the sequence diagram depicting the
process. **AW2CW mapping algorithm** is proposed here to accomplish the mapping from AW to CW.

### 5.3.1 AW2CW mapping algorithm

1. After received the mapping request from Macro-workflow modelling tool, UA sends the AW model to MA. As we know, the AW model is composed of many FMs.
2. MA sends the query about information of PAs in specific domain to IA.
3. IA returns information of all related PAs to MA.
4. MA sends out the AW bid to relative PAs based on CNP.
5. PA searches in its capability library for qualified one. Based on actual matchmaking results, PA submits the tender.
6. When the deadline is due, MA starts to evaluate all tenders for the best group of PA plans as the final substitution of the AW. After MA selects the best PA as the bid-winner, MA will sign the FM substitution contract with the PA. If all FMs in AW have their corresponding PA plans as substitutions, go to Step (8). If else, go to Step (7).
7. MA uses **Plan generating algorithm** (given in Section 5.4) to create a new PA and its new plan to satisfy the requirements of the FM. MA will sign the FM substitution contract with the new PA. Repeat this step until there are no unsatisfied FMs.
8. MA uses **Plan merging algorithm** (given in Section 5.4) to merge all corresponding plans of proper PAs, which have won the FM sub-bids. Now the final CW model can be gotten.
9. MA sends back the final CW model to the UA. The UA displays it in Macro-workflow modelling tool.
The mapping from CW to EW is actually the selecting process about proper services in SAgs. Figure 10 gives the sequence diagram depicting the process. **CW2EW mapping algorithm** is proposed here to implement the mapping from CW to EW.

### 5.3.2 CW2EW mapping algorithm

1. After received the execution command of selected CW from workflow console, UA transmits it to AdA.
2. AdA gets detailed information about the selected CW from IA. The information includes all QoS values of the related services referred in the CW. As we know, these services may have some kinds of relationship with each other. Some alterations to their QoS value should be made when these correlative services co-exit in the same workflow.
3. AdA uses GA to select the most suitable services from the candidate services set in all SAgs. Thus, we can get the final EW. The global QoS value of the workflow can be maximised using GA. GA used in service selection is depicted in our previous work [6].

### 5.4 Planning algorithms

A planning problem can be described as a three-tuple: \(< \Gamma, \Delta, \delta >\), where \(\Gamma\) denotes the initial state of the planner and \(\Delta\) denotes the set of goal states the planning system should attempt to reach, and \(\delta\) is the set of ground actions the planner can perform in attempting
to reach a goal state. A state can be represented as a conjunction of positive literals. For
equation, $At(Nanjing, \text{Frank}) \land \text{Destination(Sanya)}$ might represent a state in the travel
planning problem. A goal is a partially specified state, represented as a conjunction of
positive ground literals such as: $At(Sanya, \text{Frank}) \land \text{Hotel(PearlHotel)}$.

Plan generating algorithm, which is based on partial-order planning (POP) algorithm
[16,18], is designed to generate the agent plan. We have modified POP successor function
to permit the decomposition method used in the existing AW, and the final CW can be
gotten by Plan merging algorithm.

5.4.1 Plan generating algorithm

(1) The initial plan contains $\text{Start}$ and $\text{Finish}$, the ordering constraint $\text{Start} < \text{Finish}$,
no casual links and all the preconditions in $\text{Finish}$ are open preconditions.

(2) The successor function arbitrarily picks one open precondition $p$ on an action $B$,
and generates a successor plan for ever possible consistent way of choosing an
decision $A$ that achieve $p$. Consistency is enforced as follows:
- The causal link $A \rightarrow B$ and the ordering constrains $A < B$ are added to the plan.
  $A$ may be an existing action in the plan or a new one. If it is new, add it
to the plan and also add $\text{Start} < A$ and $A < \text{Finish}$.
- We resolve conflicts between the new causal link and all existing actions, and
  between the action $A$ (if it is new) and all existing causal links. A conflict
  between $A \rightarrow B$ and $C$ is resolved by making $C$ occur at some time outside the
  protection interval, either by adding $B < C$ or $C < A$ We add successor states
  for either or both if they result in consistent plan.

(3) The goal test checks that a plan is a solution to the original planning problem.
Because only consistent plans are generated, the goal test just needs to check that
there are no open preconditions.

5.4.2 Plan merging algorithm

(1) Select one PA $c'$ in AW and for the most suitable Decomposition $(c, d)$ method
from the plan library such that $c$ and $c'$ unify with substitution $\theta$, we replace $c'$ with
$d' = \text{SUBST}(\theta, d)$.
- First, the PA $c'$ is removed from AW. Then, for each step $s$ in the decomposition
d', we need to choose an action to fill the role of $s$ and add it to the workflow.
- Hook up the ordering constraints for $c'$ in the original workflow to the steps in $d'$.
- Hook up causal links. If $B \rightarrow c'$ was a causal link in the original workflow,
  replace it by a set of causal links from $B$ to all the steps in $d'$ with precondition $p$
  that were supplied by the $\text{Start}$ step in the decomposition $d$.

(2) Continue step (1) until all PAs in the original AW have been dealt with. Thus, we
get the final CW.

6. System implementation and case results

6.1 System prototype implementation

AGWMS is developed with Java language. A prototype has been developed according
to the framework. Figure 11(a) gives a screenshot of Macro-workflow modelling tool.
Figure 11(b) shows web service adapter interface. Our BDI multi-agent platform is developed using Jadex technology. Figure 11(c) shows the agent-hosting environment. The prototype runs on Windows platform.

6.2 Mapping procedure

Here we will describe the mapping process in the case, which has been introduced in Section 3.

MA utilises AW2CW mapping algorithm in the mapping process from AW to CW. Firstly, UA sends the AW model to MA as the response of user request. Then MA makes six FM sub-bids according to the AW model definition. In these sub-bids, some constrains are given, e.g. expected bid price, expected solution quality and expected execution time. MA will combine all sub-bids into one single AW bid. After getting all related PAs' information in the application domain, MA will send the AW bid to proper PAs.

After PA receives the AW bid, it will analyze the bid contents carefully. Based on the capabilities it owns, decisions about bidding will be made. Here, we take some PA decisions as examples:

- **Registration-A0** PA can only satisfy the requirements of **Register_Conference** sub-bid. Its capability **ConfRegistration** is selected. **Registration-A0** PA will make the sub-render aiming at single FM whole-substitution based on the capability;
- **Registration-C0** PA can satisfy none of sub-bids. It will keep silence and respond nothing;
- **Register&Booking_SP-C1** PA can satisfy not only the requirements of **Register_Conference** sub-bid but also the requirements of **Book_Ticket** sub-bid. Its capability **ConfRegistration&TranspBooking** is selected.
PA will make the sub-render aiming at multi-FM substitution based on the capability;

- Hotel Reservation-A3 PA can only satisfy the partial requirements of Reserve Accommodation sub-bid. Its capability HotelReservation is selected. Hotel Reservation-A3 PA will make the sub-render aiming at single FM part-substitution based on the capability.

When the deadline is due, MA receives many tenders from these PAs. MA will evaluate them and select proper PAs as the final contractors. The contracting results are given as below:

- Register Conference sub-bid has three bidders: Registration-A0 PA, Registration-B0 PA and Register&Booking SP-C1 PA. Their corresponding capabilities can all satisfy functionality requirements and other constrains given in the sub-bid. But Register&Booking SP-C1 PA is the bidder who submits the multi-FM substitution tender, so it will get the multi-FM substitution contract from MA according to Maximum FM substitution rule.

- Book Ticket sub-bid has two bidders: Ticket Booking PA and Register&Booking SP-C1 PA. Their corresponding capabilities can all satisfy functionality requirements and other constrains given in the sub-bid. But MA has signed the multi-FM substitution contract with Register&Booking SP-C1 PA to implement the substitutions of Register Conference FM and Book Ticket FM.

- Rent Car sub-bid has two bidders: Rental SP-A2 PA and Rental SP-B2 PA. Their corresponding capabilities can all satisfy functionality requirements and other constrains given in the sub-bid. Rental SP-A2 PA can help MA get benefits:

\[
\text{Benefits}_{\text{MA}} = 0.5 \times [(500 - 450)/500] + 0.3 \times [(600 - 500)/600] \\
+ 0.2 \times [(100 - 80)/100] = 0.14
\]

and Rental SP-B2 PA can help MA get benefits:

\[
\text{Benefits}_{\text{MA}} = 0.5 \times [(500 - 480)/500] + 0.3 \times [(600 - 540)/600] \\
+ 0.2 \times [(100 - 80)/100] = 0.09
\]

Obviously, Rental SP-A2 PA can help MA get more benefits, so MA will sign the single FM substitution contract with Rental SP-A2 PA according to Maximum MA benefit rule.

- Arrange Tour sub-bid has two bidders: Tour Arrangement-A5 PA and Tour Arrangement-B5 PA. Their corresponding capabilities can all satisfy functionality requirements and other constrains given in the sub-bid. Tour Arrangement-A5 PA can help MA get benefits:

\[
\text{Benefits}_{\text{MA}} = 0.5 \times [(300 - 250)/300] + 0.3 \times [(1800 - 1200)/1800] \\
+ 0.2 \times [(100 - 80)/100] = 0.223
\]
Tour_Arrangement-B5 PA can help MA get benefits:

\[
\text{Benefits}_{\text{MA}} = 0.5 \times \left[ \frac{300 - 200}{300} \right] + 0.3 \times \left[ \frac{1800 - 1500}{1800} \right] + 0.2 \times \left[ \frac{100 - 80}{100} \right] = 0.257
\]

Obviously, Tour_Arrangement-B5 PA can help MA get more benefits, so MA will sign the single FM substitution contract with Tour_Arrangement-B5 PA according to Maximum MA benefit rule.

- **Reserve_Accommodation** sub-bid has no PA bidder to submit sub-tender aiming at single FM whole-substitution. In this case, MA will find another way to analyse the related sub-tenders aiming at single FM part-substitution. Reserve_Accommodation sub-bid has such tenders from Hotel_Reservation-A3 PA, Hotel_Reservation-B3 PA, Restaurant_Reservation-A4 PA, Restaurant_Reservation-B4 PA and Restaurant_Reservation-C4 PA. MA finds that the PA bidding-union by Hotel_Reservation-A3 PA and Restaurant_Reservation-B4 PA is eligible to satisfy all requirements of Reserve_Accommodation sub-bid and could help MA get the most benefits:

\[
\text{Benefits}_{\text{MA}} = 0.5 \times \left[ \frac{600 - 380}{600} \right] + 0.3 \times \left[ \frac{400 - 360}{400} \right] + 0.2 \times \left[ \frac{100 - 80}{100} \right] = 0.253
\]

Finally, MA will sign the joint-bidding contract with Hotel_Reservation-A3 PA and Restaurant_Reservation-B4 PA. In fact, the functionality ‘Reservation of accommodation’ of Reserve_Accommodation FM is composed by ‘Reservation of hotel’ and ‘Reservation of restaurant’ in the domain ontology.
Buy_Souvenir sub-bid has none PA bidder. In this case, MA uses Plan generating algorithm to create a new PA Souvenir_Buying and its new plan Buying_Souvenir to satisfy the requirements of the FM (Figure 12). MA will sign the single FM substitution contract with Souvenir_Buying PA.

At last, the CW is generated by MA using Plan merging algorithm. It is depicted in Figure 13.

For the services bound in 21 SAGs of the final CW model, we assume that we could get their QoS values in advance. Just before the CW generated in the above process is selected to run, CW2EW mapping algorithm will be utilised by AdA to get the final EW, which has the maximum QoS value. Details are omitted here.

7. Conclusion and future work
In this paper, we propose a method of using BDI agents to implement service-oriented workflow mapping, and apply it to the AGWMS of ShanghaiGrid. Three types of workflow in AGWMS are denoted as AW, CW and EW. Utilisation of them further liberates model designers from sophisticated system details. It also increases the reusability of domain-specific models and improves the fault tolerance of the system. BDI agent technology is used to meet challenges from the grid context. CNP provides a very proper negotiation model for these agents. The problem of workflow mapping has been transferred to the problem of multi-agent negotiation with the help of CNP model in AGWMS. AW2CW and CW2EW mapping algorithms are also given to accomplish service-oriented workflow mapping.

In the near future, we intend to go further using BDI agent technology in AGWMS and do a practical performance evaluation to show that our system can be efficiently used in the Grid environment. We will also consider plan mining from numerous existing domain-specific workflow models.

Acknowledgements
This paper has been supported by the grand project of the Science and Technology Commission of Shanghai Municipality (No. 05dz15005), Chinese High Technology Development Plan (No. 2004AA104340), Chinese NSF Project (No. 60473092, No. 60503041), 863 Program of China (No. 2006AA01Z172) and Natural Science Foundation of Shanghai Municipality of China (Grant No. 05ZR14081).

Notes
1. Email: cao-jian@cs.sjtu.edu.cn
2. Email: li-ml@cs.sjtu.edu.cn
3. Email: tang-fl@cs.sjtu.edu.cn

References


