Enabling Virtual Organizations with an Agent-Mediated Service Framework

Baohua Shan1,2, Yanbo Han1, Hongcui Wang1,2
1 Research Centre for Grid and Service Computing, Institute of Computing Technology, Chinese Academy of Sciences
2 Graduate School of the Chinese Academy of Sciences
shanbaohua@software.ict.ac.cn; yhan@ict.ac.cn; wanghongcui@software.ict.ac.cn

Abstract

In this paper, an agent-mediated service framework is proposed to ease the construction of virtual organizations of standalone applications. Using this framework, on one hand, service providers can control the usage of their own services, and on the other hand, an agent representing certain stakeholders can dynamically search for capable partners and negotiate with their agents to dynamically form a virtual organization. Incorporating business level service composition language – VINCA into the framework further promotes the participation of business level users in constructing virtual organizations.

Keywords: Virtual organization, agent, Service, grid

1. Introduction

When solving problems, problem solvers always need to choose partners dynamically from many service providers to form a virtual organization. Problems can be solved through the collaboration between participants of a virtual organization. Virtual organization always requires integrating applications of different virtual organization members and supporting the collaboration between them. However, due to the heterogeneous characteristics of platform technologies and application development languages used by individual applications, the integration is always very difficult. Although there already have some problem-solving methods, most of them are specific to a certain environment or platform, e.g. CORBA, Java RMI and COM/DOM. Grid focuses on large-scale resource sharing, innovation application and, in some cases, high performance orientation, which provides a feasible platform for problem solving. The “real and specific” problem that underlies the Grid concept is “the coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations” [1].

However, in a virtual organization, each participant is independent and has full control over its resources, so the service provider might not want their resources be used directly by all others. Before using some services, service consumer must acquire the consent of the service provider. However, Web services are stateless and cannot support the efficient negotiation between the service consumer and the service provider. Moreover, in order to construct a virtual organization dynamically, capable partners should be found among lots of service providers. So there must be some mechanisms to support for the dynamic selection of partners. And third, in problem solving environment, non-computer professionals also need to participate in the construction of a virtual organization.

In this paper, we propose an agent-mediated service framework for the construction of virtual organization. An agent can be viewed as a software-based computer system that enjoys the following properties[2]: autonomy, social ability, reactivity and pro-activeness. The autonomy characteristic of an agent coincides with the autonomous requirements of partners in a virtual organization. Using agent for mediation, on one hand, service providers can control the usage of their own services, and on the other hand, an agent representing certain stakeholders can dynamically search for capable partners and negotiate with their agents to dynamically form a virtual organization. Business level service composition language – VINCA facilitates the participation of non-computer professionals in the construction of virtual organization.

This paper is organized as follows. In section 2, we first present the architecture of the framework and then discuss the cooperation mechanism of the framework. In section 3, we illustrate the implementation of the framework. In section 4, we use one case study in our project - AmGrid to exhibit the usage of the framework. In section 5, we give a brief conclusion.

2. Rationale and architecture

2.1. Architecture of the framework

Our framework is based on the business level service composition language we’ve developed - VINCA[3]. VINCA is a visualized and personalized
business level service composition language. Non-computer professionals can use VINCA to develop applications rapidly.

![Figure 1 Rationale of the framework](image)

As is shown in Figure 1, each service provider publishes basic information including the provider, Web services information and the agent’s information to the business collective (BC). When there are problems need to be solved, the problem solver uses VINCA to define business process which can fulfill the requirements, and then uses virtual organization definition tool to further define the virtual organization to be constructed. In the business process, the problem solver defines tasks that partners should accomplish and constraints on partners. Virtual organization management tool automatically searches the business collective for candidates according to the constraints defined by virtual organization definition tool. For each task, virtual organization agent negotiates with candidates’ agents. Depending on the strategy of negotiation, the virtual organization agent selects one partner for each task. Finally virtual organization management tool forms the virtual organization and manages the running of the virtual organization. During negotiation, the virtual organization agent also gets detail information of how to use partners’ Web services, e.g. interface definitions of the services and authentication information for using the services. The architecture of the framework is shown in Figure 2.

![Figure 2 Architecture of the framework](image)

Detail information about each module is introduced below.

2.1.1. Business Collective module

Business collective is a registry repository for service provider to register their Web services and other information. However, different from UDDI[4], the address registered is not the WSDL address of the service. It’s the address of agent located on the service provider’s side. One agent may manage several Web services. In business collective, business unit (BU) is the basic element. It is defined as follows:

**Definition 1**: Business Unit is a three-tuple: 

- `<Business Unit> ::= <Basic Information> {<Service>} <Agent>`
- `<Basic Information> ::= <Provider Name> [<Provider Description>]`
- `<Service> ::= <Service Name>{<Service Category>} {<Service Input>}{<Service Output>} {<Service NFP>}{<Service Description>}`
- `<Service Input> ::= <Name><Semantic>`
- `<Service Output> ::= <Name><Semantic>`
- `<Service NFP> ::= <Name><Semantic><Value>`
- `<Agent>::= <Name><Description><Address>`

Each service provider corresponds to a business unit in the business collective. When constructing virtual organization, virtual organization management tool uses this information to search for candidates. In each business unit, service provider can register several services but only one agent address is allowed. LSM (Lightweight Service Management) is a tool to manage Web services of the service provider. It is deployed on the service provider’s side.

2.1.2. VINCA Design and Execution module

VINCA [3] represents for A Visual and personalized Business-level Composition Language for Chaining Web-based Services. VINCA embodies an integrated approach to mediate between diverse, rapidly changing user requirements and composites of individual services scattered over the Internet. It is targeted for the application scenarios that require Web-based services be quickly assembled and coordinated by non-computer professionals to fulfill certain spontaneous requirements. The positioning of VINCA coincides with the requirements of virtual organization in that: first, business user can use VINCA to develop applications rapidly. Then problem solver can assemble relative partners to construct virtual organization quickly. Second, when VINCA process executes, it dynamically binds to concrete services and virtual organization can find appropriate partners dynamically. And third, business users can participate in the construction of virtual organization because VINCA is targeted for business users. Then we choose VINCA as the service composition language for our framework. Detailed information about VINCA can be found in[3] [10].
2.1.3. Virtual Organization module

Virtual organization module of the framework is used to construct virtual organizations and to manage them.

Virtual organization definition tool is used to define constraints of virtual organization and constraints of partners who can join the virtual organization. Virtual organization management tool uses the result produced by virtual organization definition tool to construct virtual organization. In the construction process, each virtual organization will be allocated one agent and the agent negotiates with candidates’ agents.

We define virtual organization as follows:

**Definition 2:** A virtual organization is a set of business process, constructor, constructor’s tasks, outsourcing tasks, roles, users, partners, Web services, and relations between them.

\[ \text{Virtual Organization} := \text{VO Name} \]
\[ \text{VO Description} \]
\[ \text{VO Constructor} \]
\[ \text{Business Process} \]
\[ \text{Outsourcing Task} \]
\[ \text{VO constructor Task} \]
\[ \text{Partner} \]
\[ \text{Web Services} \]
\[ \text{Role} \]
\[ \text{User} \]
\[ \text{Relation} \]

In a virtual organization, all resources (include human and services) are coordinated through a business process. Business process is composed of business services. Business services can be divided into two categories: VO constructor tasks and outsourcing tasks. A VO constructor task is a business service that can be accomplished by virtual organization constructor itself and an outsourcing task is a task that should be accomplished by partners. Each business service binds to a web service and can be executed by several roles. One role may be assigned to many users and one user may act as several roles. Elements of a virtual organization and relations between them are shown in Figure 3.

![Virtual Organization](image)

**Figure 3 Relations between Virtual Organization elements**

2.2. Cooperation mechanism

In the virtual organization construction process, first virtual organization management tool finds candidates and then virtual organization agent negotiates with candidates’ agents to determine the final partners of virtual organization. Following are the key strategies defined in our framework.

2.2.1 Search for Candidates

In order to describe the virtual organization to be constructed and constraints of partners, we’ve developed a language – VOML (Virtual Organization Markup Language). It is a XML-based language. Schema of the language is as follows. We only give the key structure of the schema:

```
<complexType name="Constraints">
  <attribute name="value" type="string"/>
  <attribute name="description" type="string"/>
</complexType>
```

Some basic concepts of the schema are defined below:

**Definition 3:** For each outsourcing task, constraints defined in VOML are satisfied only if provider restriction \( pr = \{pr_i, pr_2, \ldots, pr_n\} \), service restriction \( sr = \{sr_1, sr_2, \ldots, sr_n\} \), service provider information \( pi = \{pi_1, pi_2, \ldots, pi_n\} \) and service information \( si = \{si_1, si_2, \ldots, si_n\} \) satisfy the following rules:

(a) \( \forall pr_i \in pr \to \exists pi_j \in pi \) where \( pr_i.semantic = pi_j.semantic \) and \( pr_i.value, pi_j.value \) satisfy the operator.

(b) \( \forall sr_i \in sr \to \exists si_j \in si \) where \( sr_i.semantic = si_j.semantic \) and \( sr_i.value, si_j.value \) satisfy the operator.

**Definition 4:** There are six operators ‘>’, ‘<’, ‘=’, ‘>=’, ‘<=>’, ‘!=’ defined in our framework. \( value_i \) and \( value_j \) satisfy the operator only if:

(a) If \( value_i \) and \( value_j \) are both digits, then compare the value directly.

(b) If \( value_i \) or \( value_j \) is not digit, then treat \( value_i \) and \( value_j \) as string and check whether \( value_i \) and \( value_j \)
are equal or not. Under this circumstance, the only support operators are ‘=’ and ‘!=’.

Functionality of a service is one of the most important factors in searching for candidates. Functionality requirements of services are defined in VINCA process. We import the process definition into VOML by the ‘include’ element. Matching rules between functionality requirements and functionality provided by service provider’s service is defined as follows:

**Definition 5:** Functionality required \((fr)\) and functionality provided \((fp)\) by service provider matches only if:

(a) \(|fr.inputs| \geq |fp.inputs|, |fr.outputs| \leq |fp.outputs|\). It means that \(fp\) requires fewer inputs and produces more outputs than \(fr\).

(b) \(\forall fr.input \in fr.inputs, \exists fp.input \in fp.inputs\) where \(fr.input.semantic = fp.input.semantic\)

(c) \(\forall fr.output \in fr.outputs, \exists fp.output \in fp.outputs\) where \(fr.output.semantic = fp.output.semantic\).

(d) \(\forall fr.nfp \in fr.nfps, \exists fp.nfp \in fp.nfps\) where \(fr.nfp.semantic = fp.nfp.semantic\) and \(fr.nfp.value \subseteq fp.nfp.value\).

**Definition 6:** A service provider is considered to be a candidate only if it satisfies definition 3 and definition 5.

The outsourcing tasks defined in VOML and information that service provider has published will be matched with virtual organization management tool. If each outsourcing task of the business process has found a not empty set of candidates, then we say that the virtual organization is well defined.

**Definition 7:** A virtual organization is well defined only if for outsourcing tasks set \(ost = \{t_1, t_2, \ldots, t_n\}\), \(\forall t_i \in ost\), \(\text{Candidate}(t_i) \neq \Phi\). \(\text{Candidate}(t_i)\) represents the candidates set of task \(t_i\).

### 2.2.2. Negotiation strategy

After the phase of searching for candidates and the verification of well definition of the virtual organization, now for each task and its candidates, virtual organization agent starts the negotiation phase with the agent of each candidate. During this phase, the final partner of each outsourcing task will be determined. In this paper, we adopt the contract net protocol\[5\] to implement the negotiation between the virtual organization agent and candidates’ agents. However, we provide specific contents for our framework and give more messages. More details of our framework are given below.

First, the virtual organization agent sends task announcement message about task \(t_i\) to the agent of each candidate. The announcement message format is as follows:

**To:** Candidates’ agents  
**From:** Virtual organization agent  
**Type:** TASK ANNOUNCEMENT  
**Expiration Time:**  
**Content:**

- **Task abstraction:**  
  - **Name:** \(t_i\) (semantic)  
  - **Time:** \([t_{begin}, t_{end}]\) (semantic)

- **Inputs Provided:**  
  - **Input:**  
    - **Name:** \(i_1\) (semantic)

- **Outputs Needed:**  
  - **Output:**  
    - **Name:** \(o_1\) (semantic)

- **Bid specification:**  
  - **Time:** \([t_1', t_2']\) (semantic)

The task announcement message contains information about the outsourcing task, e.g. inputs that can be provided by the stakeholder and outputs that should be returned by the candidate’s services. And also, it contains bid information which must be provided by candidates in the bidding message. Other information is contained in the element of bid specification. All elements in the message have semantic description. When candidate’s agent receives task announcement message, it decides whether to bid for joining the virtual organization depending on the strategy defined by the service provider and services that the service provider can provide. If one of candidate’s services satisfies the requirements, then candidate’s agent may send back bid message. Bid message format is defined as follows:

**To:** Virtual Organization agent  
**From:** Candidate’s agent  
**Type:** BID  
**Content:**

- **Task abstraction:**  
  - **Name:** \(t_i\) (semantic)  
  - **Time:** \([t_{begin}, t_{end}]\) (semantic)

- **Outputs Needed:**  
  - **Output:**  
    - **Name:** \(o_1\) (semantic)

- **Bid specification:**  
  - **Time:** \([t_1', t_2']\) (semantic)

The bidding information of bid message depends on the bid specification in announcement message. For example, available time of the service that candidate can provide. If bid specification of the purchaser contains information about costs of product, then candidate’s bid message must contain cost of the product.

According to the negotiation strategy, virtual organization agent chooses one bidder and sends ‘award’ message to the candidate’s agent and ‘refuse’ message to other candidates.

**To:** Candidate’s agent  
**From:** Virtual Organization agent  
**Type:** AWARD  
**Content:**
Motivated by the above concerns and based on VINCA project, we develop a prototype of the framework.

The requirement for an agent-independent agent communication language (ACL) has led to the development of Knowledge Query and Manipulation Language (KQML)[6] and FIPA ACL[7]. In our framework, FIPA is introduced as the agent communication language. To facilitate the construction of agent, we choose ZEUS[8] as our agent generator. However, virtual organization management tool is used as the name server agent in ZUES, and some additional parts are added to parse the content of message.

We’ve developed several tools to support the framework. These tools can be divided into two categories: service provider side tools and service consumer side tools. Virtual organization construction is accomplished on the service consumer side. The tools and their functionalities are introduced below.

VOD (Virtual Organization Definition tool): virtual organization constructor can use this tool to define constrains of virtual organization, for example, define the outsourcing tasks, and for each outsourcing tasks, further define partner restrictions and service restrictions. The output of VOD is a voml file defined in section 2.2.1. VOD is a service consumer side tool.

VOM (Virtual Organization Management tool): this tool parses voml file produced by VOD tool and searches for candidates in business collective. If the virtual organization is well defined (Definition 7), then VOM provides virtual organization agent the addresses of candidates’ agents and start the negotiation phase. If virtual organization is well constructed (Definition 8), VOM manages the virtual organization for running and monitors the status of virtual organization.

LSM (Lightweight Services Management tool): this tool is deployed on service provider side and manages services that service providers have provided. Service providers also use this tool to define rules of which Web services to provide depending on the content of request message and negotiation strategy. Some other information, for example authentication information of using service is also defined in this tool.

4. Application of the framework

AmGrid[9] is a grid application platform aimed at the requirements of agile resource cooperation in network manufacturing. When new user requirements come out, the enterprise that user has submitted his requirements to will evaluate its ability of accomplishing user’s requirements. If the enterprise cannot accomplish the requirements by itself, then it will dynamically search for capable partners to construct virtual organization to fulfill user’s requirements.

Suppose that the enterprise needs to manufacture some notebooks, however, there are not enough CPU, hard disks and memories left. Then a new process...
satisfying the requirements is constructed (center of figure 4). On the left of the figure, we can see that B and C can provide 30G hard disk. If the enterprise wants to buy products from B or C directly without further negotiation, then may define B or C as the hard disk provider by clicking on B or C in figure 4. If the enterprise wants further negotiation to decide the final hard disk provider between B and C, by clicking “run virtual organization” button, virtual organization agent will negotiate with B and C’s agents to choose the final partner. During negotiation, negotiation rules of each part will be used by the agent.

Figure 4 Snapshot of virtual organization definition

5. Relate work

There already exist some methods for the construction of virtual organization in grid environment. For example CONOISE-G[11], myGrid[12] and DAME[13]. In [14], Ian Foster et al. suggest using agent technology for the construction of virtual organization because of the intelligent characteristic of agents. In myGrid and DAME, virtual organization is statically defined by using workflow, so they are incapable of handling the dynamic environment of the grid. CONOISE-G uses agent technology to cope with the dynamic environment in the grid, however, how to integrate agent and service is one of the big issues. Moreover, all these projects are at the software level and non-computer professionals cannot fully participate in the construction of virtual organization.

6. Conclusions and future work

This paper presents an agent-mediated service framework and explains in detail the definition of virtual organization, selection of candidates and negotiation strategy used between virtual organization agent and the candidate’s agent. Finally we use one case study in AmGrid to illustrate the application of the framework.

This approach has the following features:
(a) Non-computer professionals can use the tools to dynamically form an application virtual organization.
(b) Through negotiation between virtual organization agent and candidate’s agent, better partner can be attained among lots of service providers.
(c) Service providers can control the usage of their own services.

However, there still exist some important thing need to be studied, such as trust and security. In future, our work includes better algorithm for partner selection and further research on negotiation strategy.

References