Towards a Composition Framework for E-/M-Services

Zakaria Maamar
College of Information Systems
Zayed University
Po Box 19282, Dubai, U.A.E
zakaria.maamar@zu.ac.ae

Boualem Benatallah and Quan Z. Sheng
School of Computer Science & Engineering
The University of New South Wales
Sydney NSW 2052, Australia
{boualem,qsheng}@cse.unsw.edu.au

ABSTRACT

We present a framework that enables the composition of services for the benefit of users. Two types of services exist: E-services and M-services. Moreover, two types of users exist: static and mobile. The composition framework, software agents and workflows are used.

Categories and Subject Descriptors

H.4.m [**Wireless Systems**]: Services; D.2 [**System Design**]: Software agents.

General Terms

Design, Experimentation.

Keywords

Composition, E-/M-services, Wired, Wireless.

1. INTRODUCTION

With the widespread of the 3W technology, several businesses are offering their services over web portals [5]. Known as E-services, such services are meant to be triggered by users. People planning for their summer vacation can access a tourism site and specify their needs in terms of favourite airliner, accommodation type, and departure and return dates.

It is accepted that E-services have the ability to interact with other E-services in order to achieve high-level business processes. In the tourism industry, several partners are involved ranging from travel agents and airline companies to entertainment agencies. Each partner has its E-services that can invoke other E-services or be invoked as well. The collaboration of several E-services from different origins requires a composition framework. The objective is to regulate the interaction between the e-services. WorkFlow (WF) is an appropriate technology for such a framework [1].

Besides the new role of the Internet as a vehicle of delivering E-services, we witness a boom in wireless technologies. Telecom companies are offering new services to their customers over wireless devices. For instance, getting e-mails to his mobile phone is becoming natural. The next stage (if we are not already in) for Telecom is to allow users to enact E-services from their wireless devices and possibly, to run these E-services on their wireless devices.

The use of a framework that composes E-services and M-services together offers more opportunities to users to conduct operations regardless of their location and the devices they are using. E-services and M-services have to be viewed from different perspectives. First, E-services are associated with wired devices. Meanwhile, M-services are associated with wireless devices. Second, the execution of E-services occurs at the server side. Meanwhile, we advocate that the execution of M-services occurs at the client (user) side. While content delivery from fixed networks will keep increasing, it cannot be assumed that the same type or level of demand will be present in the mobile environment. The differences in network and device capability require a different solution in providing content through mobile portals. In this paper, we aim at presenting a framework that composes services regardless of their type. To this end, Software Agents (SAs) and WFs are suggested as a support technology. SAs act on users' behalf and make the composition operations transparent to users. WFs define the way SAs carry out the composition operations in terms of how to look for the relevant services, how to select the appropriate services, how to combine the services in a common business process, and finally how to implement the obtained process.

Section *2* introduces the concepts of the service composition framework. Section *3* discusses the architecture and operating of the framework. Section *4* concludes the paper.

2. BACKGROUND

An `E-service` is a software component that an organization provides in order to be assembled and re-used in a distributed, Internet-based environment [4]. A component is considered as an E-service if it is: 1) independent as much as possible from specific platforms and computing paradigms; 2) developed mainly for inter-organizational situations rather than for intra-organizational situations; and 3) easily composable; its assembling with other E-services does not require the development of complex adapters.

`M-services` should offer new opportunities to users of wireless devices. [3] considers an application component as an M-service if it is: 1) transportable through wireless networks; 2) flexible in term of composition with other M-services; 3) adaptable according to the computing features

UbiAgents Workshop'02 - AAMAS'02, July 15-19, 2002, Bologna, Italy.

of wireless devices; 4) runnable on wireless devices.

E-services and M-services are invoked for processing in two different ways: remote processing (applies only to E-services) and local processing (applies to both E-services and M-services).

3. A FRAMEWORK FOR COMPOSING E-/M-SERVICES

To motivate the E-/M-service composition framework, we consider the situation where a user is planning for his summer vacation. The user may be at home using his Internet-connected workstation or riding a bus using his mobile phone. He wants to book a domestic flight and an accommodation. He, also, also wants to find some attractions for visit. Finally, he would like to rent a car if the major attraction is far from the booked accommodation. This situation needs at least *4* services: attraction searching, flight ticketing, hotel booking, and car rental, to be connected and executed according to a specific control flow. First, the user's needs are associated with a WF. Then, services that are needed to complete these needs have to be identified. We assume the existence of several providers of services. Therefore, the use of a broker is required. It matches needs to services.

The execution of services requires computing resources (hardware and software facilities). Looking for the providers of resources for the needs of services is another step that needs to be performed. Therefore, the use of a broker is required. It matches services to resources. It happens that a provider has services but lacks the resources to perform them; he has committed all his resources to other services.

In the E-/M-service composition framework, two types of providers exist: provider of services and provider of resources. It may happen that a provider plays both roles. Based on the distinction that exists between providers, a user with a PC or mobile phone is also seen as a provider of resources. Therefore, the resources of his device have to be advertised to the broker of resources.

3.1 Architecture

Figure 1 illustrates the architecture of the system that supports the E-/M-service composition framework. The architecture consists of three parts. The first part corresponds to the providers of services (S), providers of resources (R), and providers of services and resources (both S and R). The second part corresponds to the consumers of services, i.e. users. Users have wired devices, wireless devices, or both. Finally, the third part is a meeting platform on which multiple brokers undertake the necessary matching [2]. Three categories of agents act on behalf of the components of the above-cited parts.

`Provider-agents` are specialized into two types: resource-provider-agent and service-provider-agent.

`User-agents` run on top of users' devices and are specialized into two types: wireless-user-agents for users of wireless devices, and wired-user-agents for users of wired devices.

`Broker-agents` are specialized into two types: service-broker-agent and resource-broker-agent. Service-broker-agents receive notifications from service-provider-agents regarding the services they offer. In addition, service-broker-agents receive requests from users regarding the services they need to satisfy their needs. Meanwhile, resource-broker-agents receive notification messages from resource-provider-agents regarding the resources they have. In addition, resource-broker-agents receive requests from providers of services regarding the resources they need to execute their services. Both broker-agents interact together based on the fact that services need resources on which they are completed.

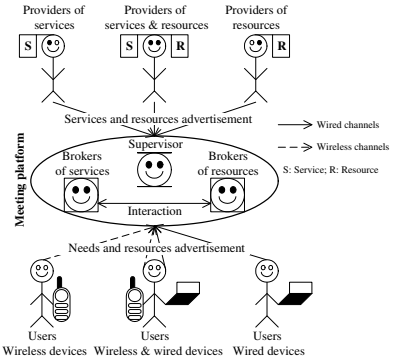

Figure 1: Framework architecture

In Figure 1, a supervisor-agent is responsible of the meeting platform. This is a common place in which agents meet and interact locally [2]. Service-broker-agents and resource-broker-agents are among the agents that interact in the meeting platform. The advertisement of services and resources are sent directly to broker-agents. Meanwhile, users' needs are sent to the supervisor-agent that assigns them to user-delegates. The supervisor-agent has a bank from which user-delegates are created (delegates are agents, but for the sake of simplicity we use the word delegates).

The presence of delegates is motivated by the fact that resources of wireless devices are limited, if they are compared to their counterpart of wired devices. Once the broker of services matches needs to services, the next stage consists of designing the WF of services. This stage needs computing resources. Since the resources of wireless devices are limited, thus they have to be used in a "rationale" way. It is more appropriate that the design of the WF of services takes place in the meeting platform instead of taking place in a wireless device. Users of wired devices can also design their WF of services in the meeting platform, if they wish. Details on how a WF of services is designed are given in the next section. In our project, computing a WF in the meeting platform is privileged.

3.2 Operating

The operating of the service composition framework goes through 6 stages. Due to lack of space, certain stages are briefly presented.

`1) Initialization:` the purpose is to undertake the "agentification" of the participants of the service composition framework.

`2) Advertisement:` the purpose is to inform brokers about the services and resources that are available. The needs of users are only advertised in the next stage. The advertisement of services concerns only providers. Service-provider-agents create service-delegates and transfer them to the meeting platform. For the advertisement of resources, it concerns both providers and users. After the resource-provider-agents create resource-delegates, they are shipped to the meeting platform arriving from provider sites and user sites. Because the computing resources on wireless devices are limited, the supervisor-agent is in charge of creat-

ing resource-delegates that identify the users of such devices. The supervisor-agent checks service-delegates and resource-delegates before both get into the meeting platform.

`3) Search for services:` the purpose is search for the services that satisfy users' needs of users. User-delegates trigger the matching process between needs and services.

An user-agent supports a user in defining his needs and submitting them to the supervisor-agent. When the needs are received, the supervisor-agent creates a user-delegate to satisfy them. For a user-delegate, the first step consists of searching for the needed services. To this end, it submits a request of services to the service-broker-agent. This latter matches needs to the services of service-delegates, using the repository of services. In case there is a match, the service-broker-agent notifies the user-delegate and service-delegates.

Despite the fact that multiple service-delegates may offer the same services, the user-delegate, at this stage, does not select any specific service-delegate. In fact, the user-delegate needs additional information before it makes a decision about the service-delegates it will be considering. The cost and version of a service are among the missing information. The outcome of `search for services` stage is a WF of the services that are needed to satisfy the user's needs (cf. Equ. (1)). $WF_{Need}^{Service} =$
$\{(service_1, \prec service\_delegate_1, service\_delegate_2,$
$service\_delegate_3 \succ), (service_2, \prec service\_delegate_1 \succ),$
$(service_3, \prec service\_delegate_2, service\_delegate_3 \succ)\}$ (1).

According to Equ. (1), the service-broker-agent responds to the user-delegate's request with $service_1$, $service_2$, and $service_3$. For example, $service_1$ is provided at the same time by $service\_delegate_{1,2,3}$. At this stage of the framework operating, the available versions (i.e. E-service or M-service) and types of processing (i.e. remote processing (RP) or local processing (LP)) of a service are not considered.

`4) Search for resources:` the purpose is to search for the resources on which the previously-identified services will be executed. Service-delegates are the trigger of the matching process between services and resources. Service-delegates conduct their search based on certain information the user-delegate will be providing to them. These information are discussed below. Furthermore, the search for resources is conducted in a sequential way, i.e. service per service. The selection of any service depends on the version and type of processing of its direct previous service.

When the user-delegate receives a response from the broker-service-agent (cf. Equ. 1), it interacts with the service-delegates that offer the first service ($service_1, \prec service\_delegate_1$, $service\_delegate_2, service\_delegate_3 \succ$). The user-delegate requests a quotation for $service_1$ that should consider the version and type of processing of that service. The service-delegates, namely $service\_delegate_{1,2,3}$ send their requests of resources to the resource-broker-agent. The purpose is to identify the appropriate resource-delegates, using the repository of resources. We recall that several resource-delegates may offer the same resources.

When the broker-resource-delegate identifies the resource-delegates, they start interacting with service-delegates. Each resource-delegate submits to a service-delegate three costs per service. The cost is calculated based on the version and type of processing of the service. When the value of cost is null, this means that the resource-delegate does not support the execution of a service according to a specific version and specific type of processing.

Since $service\text{-}delegate_2$ and $service\text{-}delegate_3$ submit to the broker-resource-agent the same request as $service\text{-}delegate_1$ did about $service_1$, both service-delegates receive the same responses (cf. Equ. (2)).
$Service\_delegate_1 : Service_1 =$
$\{\prec resource\_delegate_1,$
$cost_{e-service_1}^{LP}, cost_{e-service_1}^{RP}, cost_{m-service_1}^{LP} \succ$
$\prec resource\_delegate_2,$
$cost_{e-service_1}^{LP}, null, cost_{m-service_1}^{LP} \succ\}$ (2)

According to Equ. (2), $service\_delegate_1$ gets 6 offers of resources for $service_1$; 3 from $resource\_delegate_1$ and 3 from $resource\_delegate_2$. An offer is interpreted as follows. $Resource\_delegate_1$ supports the execution of both versions of $service_1$ (E-, M-). Furthermore, $resource\_delegate_1$ supports the three types of processing, two for the e-version (R, L) and one for the m-version (L). For each version and type of processing of $service_1$, a separate cost is established.

After all the service-delegates receive responses from resource-delegates, they select the "best" resource-delegate. Each service-delegate may have its own selection strategy. At this time of our research, the selection is limited to minimizing the cost of a service, considering its version and type of processing. The following equations are obtained:
$Service\_delegate_1 : Service_1,$
$\prec resource\_delegate_1, cost_{e-service_1}^{LP} \succ$
$Service\_delegate_2 : Service_1,$
$\prec resource\_delegate_1, cost_{e-service_1}^{LP} \succ$
$Service\_delegate_3 : Service_1,$
$\prec resource\_delegate_1, cost_{e-service_1}^{LP} \succ$ (3)

At this stage of the framework operating, each service-delegate knows exactly for the `first` service the version to be included and type of processing to be used. Two significant elements are obtained from Equ. (3): 1) `one` service-delegate from the list of $service\_delegates_{1,2,3}$ has to transfer the e-version of $service_1$ to the site of $resource\_delegate_1$, and 2) the user_delegate has to migrate to the site of $resource\_delegate_1$ so it can process $service_1$ locally.

`5) Refinement:` the purpose is to refine the details that were established in search for services stage regarding the first service. Using the outcome of `search for resources` phase, Equ. (3) is integrated into Equ. (1). Since a service-delegate knows currently the version and type of processing of a service it has selected, it offers to the user-delegate a cost for that service according to that version and type of processing. In its offer, the service-delegate considers the cost of running the service on a resource. Equ. (4) illustrates the refinement of the WF/first service, using the responses of service-delegates to the user-delegate. Note: del. stands for delegate; res. stands for resource; ser. stands for service
$WF_{Need}^{Service} : Service_1 =$
$\{\prec ser.\_del._1, cost_{1:e-service_1}^{LP} \succ / [res.\_del._1, cost_{e-service_1}^{LP}]$
$\prec serv.\_del._2, cost_{2:e-service_1}^{LP} \succ / [res.\_del._1, cost_{e-service_1}^{LP}]$
$\prec serv.\_del._3, cost_{3:e-service_1}^{LP} \succ / [res.\_del._1, cost_{e-service_1}^{LP}]\}$ (4)

The user-delegate selects for the first service a particular service-delegate using the cost criterium. It minimizes the cost of getting a service from service-delegates. Equ. (5) illustrates the final definition for the first service.
$WF_{Need}^{Service} : (Service_1,$
$\prec ser.\_del._1, cost_{1:e-service_1}^{LP} \succ / [res.\_del._1, cost_{e-service_1}^{LP}])$ (5)

When the user-delegate selects a service-delegate, this service_delegate notifies the resource-provider-agent, through the resource-delegate, about the service it will receive for execution, the version of this service, who is going to pro-

cess that, and how this service will be processed.

`6) Completion`: search for services, search for resources, and refinement stages were dedicated to the first service of the WF to set up. In `completion` stage, the focus now is on the remaining services$_{i,\ i \neq 1}$ (i.e. service$_2$ and service$_3$ of Equ. (1)). The selection of any service$_i$ depends directly on the version and type of processing of service$_{i-1}$. In addition, two selection criteria are involved: cost and location. The user-delegate privileges location to cost. `Location` criterium aims at getting the maximum number of services to be executed in the same site. Site defines the locations of resource-delegates and user-delegate. `Cost` criterium aims at minimizing the cost of getting a service from service-delegates. After the selection of the first service is completed (cf. Equ. (5)), the user-delegate asks all the service-delegates of the next service to start searching for the resource-delegates according to what follows.

$\forall\ i,\ i > 1$, service$_{i-1}$ and service$_i$ are used for illustration purposes. We assume that service_delegate$_{i-1}$ provides service$_{i-1}$ and resource_delegate$_{i-1}$ supports the execution of service$_{i-1}$ in site$_{i-1}$. The selection of service$_i$ depends on: 1) the version and type of processing of service$_{i-1}$; 2) the execution location of service$_{i-1}$ (this location corresponds to the site of the resource-delegate); and 3) the current location of the user-delegate.

In what follows, the approach that is adopted to work on the definition of service$_i$ is discussed. Three cases are considered, namely Service$_{i-1}$: e-version and remote processing, Service$_{i-1}$: e-version and local processing, and Service$_{i-1}$: m-version and remote processing. Due to lack of space, only the first case is described.

`Service`$_{i-1}$`: e-version and remote processing`

Since the execution of e-service$_{i-1}$ is done remotely, this means that the user-delegate is in any site$_j$ where $j \neq (i-1)$. Three exclusive situations are available to the user-delegate to make a decision about service$_i$.

`a)` The execution of service$_i$ takes place in the same site as service$_{i-1}$, according to the location criterium. Therefore, the user-delegate requests from service-delegates that offer service$_i$ to check with the broker-resource-agent if resource-delegate$_{i-1}$, also, supports the remote processing of e-service$_i$. If resource-delegate$_{i-1}$ supports the remote processing of e-service$_i$, then the service-delegates select that resource-delegate. Afterwards, the user-delegate chooses a service-delegate based on the cost criterium (cf. Equ. (4)). This means that e-service$_i$ and e-service$_{i-1}$ will be both executed in site$_{i-1}$. The user-delegate will be invoking them remotely. Equ. (6) extends Equ. (5) after the selection of service$_2$.

$$WF_{Need}^{Service} =$$
$$\{(Service_1,$$
$$\prec ser.\_del._1, cost_{1:e-service_1}^{RP} \succ / [res.\_del._1, cost_{e-service_1}^{RP}])$$
$$(Service_2,$$
$$\prec ser.\_del._?, cost_{?:e-service_2}^{RP} \succ / [res.\_del._1, cost_{e-service_2}^{RP}])\} \quad (6)$$

`b)` The execution of service$_i$ takes place in the same site, i.e. site$_j$, where the user-delegate is currently running, according to the location criterium. This situation happens when resource_delegate$_{i-1}$ does not support the remote processing of e-service$_i$. Therefore, the user-delegate requests from service-delegates that offer service$_i$ to check with the resource-broker-agent if resource-delegate$_j$ of site$_j$ also supports the local execution of service$_i$, either as an e-version of m-version. If resource-delegate$_j$ supports the local processing of service$_i$, then the service-delegates select that resource-delegate. Afterwards, the user-delegate chooses a service-delegate based on the cost criterium (cf. Equ. (4)). This means that e-service$_{i-1}$ and service$_i$ will be executed in two different sites. However, user-delegate and service$_i$ will be in the same site. Equ. (7) extends Equ. (5) after the selection of service$_2$.

$$WF_{Need}^{Service} =$$
$$\{(Service_1,$$
$$\prec ser.\_del._1, cost_{1:e-service_1}^{RP} \succ / [res.\_del._1, cost_{e-service_1}^{RP}])$$
$$(Service_2,$$
$$\prec ser.\_del._?, cost_{?:?-service_2}^{LP} \succ / [res.\_del._j, cost_{?-service_2}^{LP}])\} \quad (7)$$

`c)` The execution of service$_i$ takes place in any site$_k$ where $k \neq (i-1)$ and $k \neq j$. This situation takes place when resource_delegate$_j$ does not support the local processing of service$_i$. In that case, the location criterium is not considered. Search for services and search for resources stages are carried out in order to define the version and type of processing of service$_i$ and the respective resource-delegate. Equ. (8) extends Equ. (5) after the selection of service$_2$.

$$WF_{Need}^{Service} =$$
$$\{(Service_1,$$
$$\prec ser.\_del._1, cost_{1:e-service_1}^{RP} \succ / [res.\_del._1, cost_{e-service_1}^{RP}])$$
$$(Service_2,$$
$$\prec ser.\_del._?, cost_{?:?-service_2}^{?P} \succ / [res.\_del._?, cost_{?-service_2}^{?P}])\} \quad (8)$$

When the selection process is completed for all the services, Equ. (9) illustrates a sample of the final outcome that can be obtained.

$$WF_{Need}^{Service} =$$
$$\{(service_1,$$
$$\prec ser.\_del._2, cost_{e-service_1}^{RP} \succ / [res.\_del._1, cost_{e-service_1}^{RP}])$$
$$(service_2,$$
$$\prec ser.\_del._1, cost_{m-service_2}^{LP} \succ / [res.\_del._3, cost_{m-service_2}^{LP}])$$
$$(service_3,$$
$$\prec ser.\_del._3, cost_{m-service_3}^{LP} \succ / [res.\_del._3, cost_{m-service_3}^{LP}])\} \quad (9)$$

4. CONCLUSION

In this paper, we presented an overview of our research on composing services, regardless of their types (i.e. E-service or M-service). The composition framework relies on two technologies: SAs and WFs. SAs carry out composition operations on behalf of users. And, WFs specify to SAs how the composition occurs. Our aim is to allow users to satisfy their needs regardless of their location and the devices they are using. Several issues need to be tackled, among them how to specify the WF that connects services together, and how to deal with the limitations of wireless devices.

5. REFERENCES

[1] D. Georgakopoulos, M. Hornick, and A. Sheth. An overview of workflow management: From process modeling to workflow automation infrastructure. *Distributed and Parallel Databases*, 3:119–153, 1995.

[2] Z. Maamar, E. Dorion, and C. Daigle. Towards virtual marketplaces for e-commerce. *CACM*, 44(12), Dec. 2001.

[3] Z. Maamar, W. Mansoor, and Q. H. Mahmoud. Software agents to support mobile services. In *Pro. of the 1st Int. Joint Conference on AAMAS (Poster Session)*, Bologna, Italy, 2002.

[4] B. Pernici and M. Mecella. Designing components for e-services. In *Pro. of the VLDB Workshop on Technologies for E-Services*, Cairo, Egypt, 2000.

[5] J. Yang, M. Papazoglou, and W.-J. van den Heuvel. Tackling the challenges of service composition in e-marketplace. In *Pro. of the 12th RIDE-2EC'2002 in conjunction with ICDE'02*, San Jose, USA, 2002.