Towards a Unified Framework for Composing E-/M-Services

Zakaria Maamar[1], Boualem Benatallah[2], and Quan Z. Sheng[2]

[1] College of Information Systems
Zayed University
PO Box 19282, Dubai, UAE
zakaria.maamar@zu.ac.ae

[2] School of Computer Science & Engineering
The University of New South Wales
Sydney NSW 2052, Australia
{boualem,qsheng}@cse.unsw.edu.au

Abstract This paper discusses an approach to design a framework that enables the composition of services for the benefit of users. Services are decomposed into two types: e-services and m-services. Similar to services, users are decomposed into two types: static and mobile. To set up the composition framework, the approach consists of the following steps: understand the features of wireless devices in term of computing, storage, and display capabilities; define the characteristics of e-services *vs.* m-services; define the principles of the framework that composes e-services and m-services; suggest selection criteria between an e-service and m-service; and finally implement the framework using a running example, e.g. travel planning.
Keywords: e-services, m-services, composition, software agents.

1 Introduction

It is widely accepted that e-services have the ability to engage other e-services in order to complete high-level business processes. In tourism industry, several partners are involved ranging from travel agents and airline companies to entertainment agencies [7]. Each partner has its e-services that can invoke other e-services or can be invoked by other ones. However, requesting the collaboration of several e-services from different origins requires a composition framework. Workflows (WF) technology is one of the appropriate technologies that support the design of such a framework. By e-service (also called Web service), we mean a semantically well defined abstraction that allows users to access functionalities offered by Web applications [1]. A typical example of a e-service is booking an airline ticket through an HTML-based interface.

Besides the new role of the Internet as a vehicle of delivering e-services, we all witness a boom in wireless technologies [10]. Telecom companies are offering new opportunities to users over wireless devices (e.g., mobile phones). Reading emails and sending SMS messages between mobile phones are becoming popular. Surfing the web thanks to the Wireless Application Protocol (WAP) is another evidence of the wireless technology expansion. The next stage for telecom companies in collaboration with businesses is to allow users to enact e-services from their wireless devices and possibly, to run these e-services on their devices[1]. M-services denote this type of e-services. Throughput, connectivity, and bandwidth of wireless networks are in constant improvement [8,13].

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. The rationale of designing such a framework is to shed the light on the fact that 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. On the other hand, we advocate that the execution of M-services should occur at the user side. [8] argues that 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 discussing a framework that composes services regardless of their types. To this end, Software Agents (SAs) [4] and WFs are suggested as the support technologies. SAs act on behalf of users 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, and how to combine the services in a common framework.

The paper is organized as follows. Section 2 gives the background of software agents, workflows, and services. Section 3 presents the service composition framework. The types of agents used in the proposed framework are presented in Section 4 and conclusion is given in Section 5.

2 Background

Software agents (SA). An SA is a piece of software that acts to accomplish tasks on behalf of user [4]. The design of most SAs is based on the approach that the user needs only to specify a high-level goal instead of issuing explicit instructions, and leave the "how" and "when'" decisions to the agent. An SA exhibits a number of features that make it different from other traditional components: autonomous, goal-oriented, collaborative, flexible, self-starting, temporal continuity, character, communicative, adaptive, and mobile.

Workflows. The Workflow Management Coalition defines a WF as "the automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for actions, according to a set of procedural rules".

Mobile computing. Mobile computing refers to systems in which computational components, either hardware or software, change locations in a physical environment [14]. Moving from one location to another is due to several reasons: components miniaturization, new communication channels, and last but

[1] We consider triggering an e-service from a wired (wireless) device as a weak definition to be associated with an E-service (M-service).

not least mobile-code programming languages. Samples of mobile computing include [14]: hardware mobility (e.g., notebook computers), software mobility (e.g., Java applets), and combined mobility (e.g., Java enabled I-mode applications http://www.nttdocomo.co.jp/english/p_s/imode/index.html).

E-services vs. *M-services* Regardless of its type, we consider a service as a procedure that completes specific operations according to specific inputs. Potential users of a service have to know how to invoke a service but do not have to know how to operate the service or even how the service operates or is operated.

We consider the difference between an E-service and M-service at two different levels. The first level concerns the communication medium (i.e., wired channel *vs.* wireless channel). The second level concerns the location of where the processing occurs: server side (Can be client side as represented by Figure 2-b-) on a wired platform for an E-service *vs.* user side on a wireless platform for an M-service. According to Figure 1, E-services and M-services are invoked in different ways.

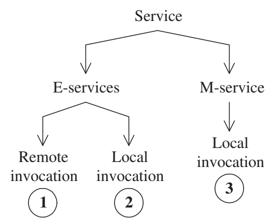

Figure1. E-service invocation *vs.* M-service invocation

1. Remote invocation (only applies to E-services): the client sends remotely a request to the server asking the execution of a service. The execution takes place in the server. During the execution, a permanent and stable communication channel is established between the client and the server. In the remote invocation, the server performs a security step to verify the client's identity before initiating any service.
2. Local invocation (applies to both E-service and M-service): the client asks remotely the server to transfer a copy of a service to its site. After being transferred, the execution of the service takes place in the client. A security step is carried out in the local invocation. The purpose is twofold: the server checks the client's identity before sending any service, and the client checks the service's security credentials upon reception.

A local invocation of a service can be viewed from two different perspectives [6]. The server perspective occurs when a client is able to move from one site to another (i.e. the client is mobile, see Figure 2-a-). When the client arrives to the server site, it invokes the service locally. The client perspective occurs when the client receives a copy of a service from the server and then, runs the service in its site (see Figure 2-b-). In the client perspective, it is assumed that the service is self-contained; the service does not require any extra resources from the server to which it belongs.

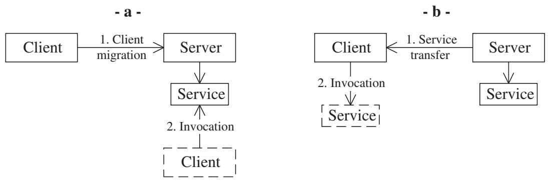

Figure2. Types of local invocation of a service

3 Composing E-/M-services

3.1 Motivation example

To motivate our work on the E-/M-services composition framework, we consider a trip planning example: (i) a user wants to book a domestic flight, as well as an accommodation; (ii) he also wants to find some attractions for visit; (iii) he would like to rent a car if the major attraction is far from the booked accommodation.

The framework should help the user in meeting his requirements. First, the user's needs are converted into a request associated with a WF. Once the WF is decomposed into tasks (based on specific decomposition templates), the corresponding services that are needed to complete these tasks are located. At this stage, the use of a broker is required. Using a repository of meta-data that describes services, the broker matches tasks to services (see Figure 3 (a)).

The execution of services requires computing resources. By resources, we mean hardware and software means. Looking for the providers of resources for the benefit of services is another step that needs to be performed. Similarly, a broker is also needed at this stage. Using a repository of meta-data that describes resources, the broker matches services to resources (see Figure 3 (b)). It happens that a provider has services but lacks the resources to perform them (e.g., the provider has committed all available resources to other services). Therefore, two types of providers are distinguished: provider of services and provider of resources. A provider can play both roles. A user with a PC or mobile phone is also considered as a provider of resources. The resources of his device can be first advertised and then, requested for executing services. Considering users as providers of resources is an important research avenue that does not limit the role of users to just consumers; their resources may be involved in satisfying the needs of other users.

3.2 Architecture

The design and development of a composition framework of services (regardless of the type of services) requires the identification of the entities involved in such a framework. Several studies agree on three entities [11,12]: *Service providers*, *Service requesters*, and *Broker's registry*. Service providers create and define services

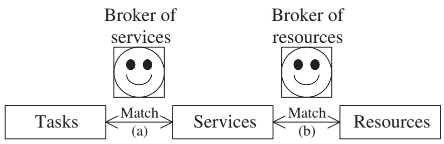

Figure3. Tasks *vs.* Services & Services *vs.* Resources

using languages, such as the Web Services Description Language (WSDL). For advertisement purposes, providers register their services with brokers. A broker maintains a registry of advertised services. The Universal Description, Discovery, and Integration (UDDI) protocol offers a standard way for services to publish themselves with registry and discover each other. The Simple Object Access Protocol (SOAP) provides an XML-based mechanism for exchanging structured data between network applications.

Figure 4 illustrates the general architecture of the framework that we propose for composing services. The architecture consists of three parts: *provider*, *user*, and *broker*. The provider part corresponds to the providers of services (S), providers of resources (R), and providers of services and resources (both S and R). The user part corresponds to the consumers of services. Users are associated with wired devices, wireless devices, or both. The broker part is a meeting platform on which multiple brokers undertake the necessary matching operations [5]. The platform is the link between the provider part and user part. Each part of the architecture of Figure 4 has several types of agents that act upon the components of the above-cited parts, namely provider, user, and broker. Three categories of agents exist: (i) Provider-agents are specialized into two types: resource-provider-agent and service-provider-agent; (ii) User-agents are specialized into two types: wireless-user-agents and wired-user-agents; (iii) Broker-agents are specialized into two types: broker-service-agent and broker-resource-agent. Details of the agents are presented in Section 4.

In Figure 4, a supervisor-agent manages the meeting platform. The meeting platform is a common place in which agents meet and interact locally. Broker-service-agents and broker-resource-agents are among the agents that interact in the meeting platform. The advertisement of services and resources are sent directly to broker-agents, assuming that the contact addresses of these brokers are known to resource-provider-agents and service-provider-agents. Meanwhile, users' needs are sent to the supervisor-agent that assigns them later on to user-delegates. The supervisor-agent is in charge of a bank from which user-delegates are created (delegates are agents, but for the sake of simplicity we use the word delegates. We advocate that each user will be associated with a user-delegate in the meeting platform. User-delegates are in constant interaction with user-agents.

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 sequence of services to be invoked. The performance of this

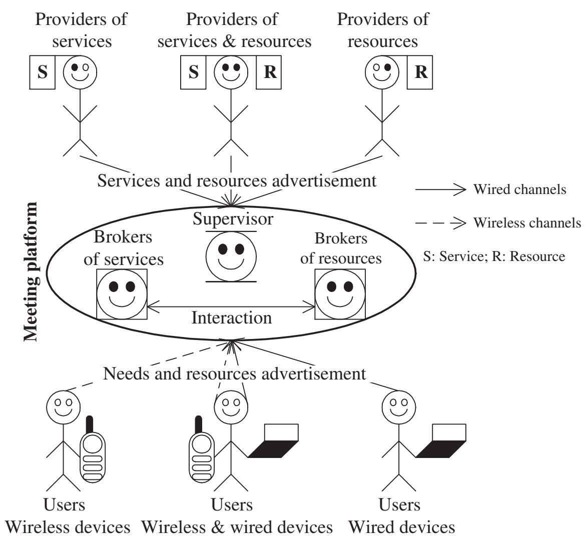

Figure4. Agent-based architecture of the service composition framework

stage needs computing resources. It is largely accepted that resources of wireless devices have to be used in a "rationale" way. Therefore, it is more appropriate that the design of the sequence of services takes place in the meeting platform instead of wireless devices. Users of wired devices can also design their sequence of services in the meeting platform, if they wish. In addition, a user who has a wired device and decides to submit his needs from a wireless device to the supervisor-agent has the opportunity to design the sequence either in the meeting platform or in his wired device. The way a sequence of services is designed is part of our current work. We privilege the option of computing a sequence in the meeting platform. Another argument in favor of the meeting platform concerns a user-delegate that may need additional information to complete the design of a sequence of services. Since the user-delegate is already in the meeting platform, it interacts directly with the relevant broker. There is no need to send remote requests.

Figure 5 summarizes the approach user-delegates are created. Once $\texttt{User}_1$ submits his needs to the supervisor-agent, $\texttt{User-delegate}_1$ is installed in the meeting platform (see Figure 5 a). Meanwhile, after $\texttt{User}_2$ submits his needs to the supervisor-agent he has the opportunity to select where the installation of $\texttt{User-delegate}_2$ takes place, either in the meeting platform (see Figure 5 a) or in his wired device (see Figure 5 b). As we stated in the previous paragraph, we privilege the meeting platform.

There are two types of advertisements in Figure 4. The first type of advertisement corresponds to the services and resources that the providers respectively offer and have. The description of services and resources is based on an XML representation of the meta-data that are associated with either the services or resources. Samples of the meta-data are:

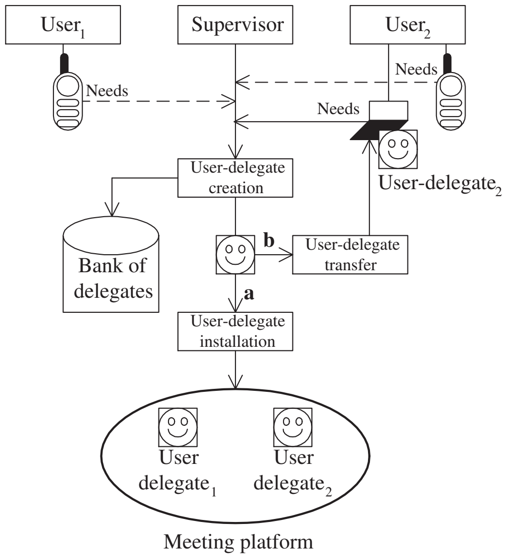

Figure5. Approach of user-delegates creation

Service:	**Resource:**
service-identifier,	resource-identifier,
type (e-service or m-service),	type (hardware or software),
description, provider-identifier,	description,
input-parameters, output-parameters,	provider-identifier,
cost, needed-resources (OS, min RAM,	type-of-provider (user or regular provider),
min CPU, min bandwidth for transfer).	cost.

The second type of advertisement corresponds to the needs and resources that the users have. Similar to the first type of advertisement, the description of needs and resources is based on an XML representation of the meta-data that are associated with the needs and resources. The meta-data of resources are above presented. The meta-data of the needs depend on the application domain. For travel planning as a domain, the following meta-data can be used: place of departure, place of destination, date of departure, date of return, number of persons, and accommodation type.

4 Types of agents

The service composition framework has seven types of agents: supervisor-agent, resource-provider-agent, service-provider-agent, wireless-user-agent, wired-user-agent, broker-service-agent, and broker-resource-agent. Moreover, three types of delegates are added to that given number of agents. These delegates are: user-delegate, service-delegate, and resource-delegate. The rationale of user-delegates has been explained in Section 3. Meanwhile, the rationale of service-delegates and resource-delegates is discussed in the current section.

Resource-provider-agent. It acts on behalf of provider of resources. Thus, it achieves the autonomy and independence of such resources. The resource-provider-agent advertises its resources whether hardware or software at the level of the broker-resource-agent. These advertisements are meant to service-provider-agents that need resources to complete their services. Once appropriate resources are identified, resource-provider-agenta receive messages from provider-service-agents requesting the use of their resources.

Service-provider-agent. It acts on behalf of provider of services. Services are required to satisfy users' needs. Therefore, they are advertised at the level of the broker-service-agent. These advertisements are meant to wired/wireless-user-agents that need services to satisfy their users' needs. Services are offered in two different versions: E-version and M-version. Each version has specific computing resources to be offered by resource-provider-agents.

Wireless-user-agent. It acts on behalf of user who is associated with a wireless device. The wireless-user-agent receives the user's needs and transfers them to the supervisor-agent of the meeting platform. The wireless-user-agent is also in charge of monitoring the services that will be received for execution on the user's wireless device. These services are of type M-services (see Figure 1). The wireless-user-agent is also aware of the features of the wireless device in terms of display size, operating system, input type, memory, storage space, and processing power. These features need to be known due to the computing requirements of M-services.

Wired-user-agent. It acts on behalf of user who is associated with a wired device. The wired-user-agent receives the user's needs and transfers them to the supervisor-agent of the meeting platform. In addition, the wired-user-agent is in charge of monitoring the services that are received for execution on the user's wired device. These services are of type E-services (see Figure 2 b).

Broker-service-agent. Its main role is to match the needs of users to the services of providers of services. The broker-service-agent receives advertisements from service-provider-agents regarding their offers of services and from wired/wireless-user-agents (through their user-delegates) regarding their needs of services. In case there is a match, the broker-service-agent notifies locally the user-delegate and remotely the service-provider-agents. Since remote interactions are always subject to different obstacles such as network reliability and transfer safety, providers of services are associated with service-delegates to be shipped from the sites of the providers of services to the meeting platform. Once the broker-service-agent informs the service-delegate about the services that match certain needs, the service-delegate notifies first, its service-provider-agent and then, the broker-resource-agent about the resources it needs to complete the requested services.

Broker-resource-agent. Its main role is to match the services of providers of services to the resources of providers of resources. The broker-resource-agent receives advertisements from service-provider-agents regarding their needs of resources and from resource-provider-agents regarding their offers of resources. The service-delegate is in charge of initiating the matching process. In case there is a match, the service-delegate has to interact remotely with resource-provider-agents regarding the modalities of using their resources. These interactions can take place locally in the meeting platform. To this end, resource-provider-agents create resource-delegates and ship them to the meeting platform.

Supervisor-agent. It is in charge of managing the content of the meeting platform. First, it creates user-delegates following the messages of needs to satisfy it receives from wireless/wired-user-agents. Each user is associated with one user-delegate. Second, the supervisor-agent checks the service-delegates and resource-delegates once they arrive from the sites of the providers to the meeting platform. Last but not least, the supervisor-agent monitors the computing resources that are used by user-delegates and service-delegates. User-delegates work on the sequence of services to be invoked to satisfy users' needs. Meanwhile, service-delegates work on identifying the resources to be used.

User-delegate. It acts on behalf of wireless/wired-user-agent in the meeting platform. The supervisor-agent creates user-delegates and assigns them to users (see Figure 5). User-delegates are responsible of defining the sequence of services to be used.

Service-delegate. It acts on behalf of service-provider-agent in the meeting platform. This agent creates the delegate and ships it to the platform. The service-delegate notifies the service-provider-agent about the agreements it has with user-delegates. Furthermore, the service-delegate advertises services to the broker-service-agent.

Resource-delegate. It acts on behalf of resource-provider-agent in the meeting platform. This agent creates the delegate and ships it to the platform. The resource-delegate notifies the service-provider-agent about the agreements it has with service-delegates. Furthermore, the resource-delegate advertises resources to the broker-resource-agent.

Figure 6 represents the delegates that act on behalf of the different agents in the meeting platform. Two types of brokering exist; the first one is related to services and the second one is related to resources. The service-delegate is the common component between the two brokering systems. When a user-delegate accepts the services of a service-delegate, this latter has to look for the resources to be used by the services for their execution.

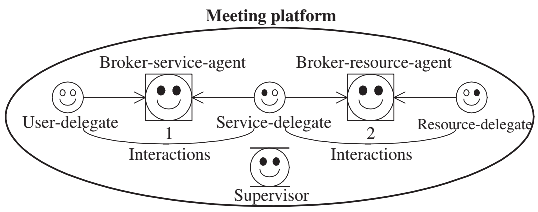

Figure6. Delegates of the service composition framework

5 Conclusion

In this paper, we presented an overview of our work on composing services, regardless of their types (i.e. E-service or M-service). The composition framework relies on two technologies, namely 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.

References

1. Boualem Benatallah, Marlon Dumas, Quan Z. Sheng, and Anne H.H. Ngu. Declarative Composition and Peer-to-Peer Provisioning of Dynamic Web Services. In *Proc. of 18th Int. Conference on Data Engineering, IEEE Computer Society*, pages 297–308, San Jose, USA, 2002.
2. N. Jennings, K. Sycara, and M. Wooldridge. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, 1(1):7–38, 1998.
3. Z. Maamar, E. Dorion, and C. Daigle. Towards virtual marketplaces for e-commerce. *Communications of the ACM*, 44(12), December 2001.
4. Z. Maamar. Moving code (servlet strategy) vs. inviting code (applet strategy) - which strategy to suggest to software agents? In *Proceedings of the 3rd International Conference on Enterprise Information Systems (ICEIS'2001)*, Setubal, Portugal, 2001.
5. P. O'Brien. Dynamic travel itinerary management: the ubiquitous travel agent. In *Proceedings of the Australian Conference on Information Systems*, Coffs Harbour, Australia, 2001.
6. D. Ralph and C. G. Shephard. Services via mobility portals. *BT Technology Journal*, 19(1), January 2001.
7. K. Rindtorff and M. Welsch. Pervasive and mobile computing middleware. *Informatik/Informatique*, 5, 2001.
8. J. Roy and A. Ramanujan. Understanding web services. *IEEE IT Professional*, November/December, 2001.
9. S. J. Vaughas-Nicholas. Web services: beyond the hype. *IEEE Computer*, 35(2), February 2002.
10. R. Vingralek. Supporting e-commerce in wireless networks. Technical report, INTERTRUST STAR*LAB Technical Report STR-TR-01-07, InterTrust Technologies Corporation, California, USA, 2001.
11. A. I. Wand and L. Chunnian. Process support for mobile work across heterogeneous systems. Technical report, Department of Information Science, Norwegian University of Science and Technology, Norway, 2001.