An Agent-based Approach to Specify a Web Service-oriented Environment

Zakaria Maamar, Fahim Akhter, and Mohammed Lahkim
College of Information Systems, Zayed University, Dubai, U.A.E
{zakaria.maamar,fahim.akhter,mohammed.lahkim}@zu.ac.ae

Introduction

With the rapid development of information technologies, several businesses are in the process of setting up web portals. The main objective is to offer their Web services thus, a large community of users can browse, select, and trigger to satisfy their needs. Also known as E-services [1], services[1] can be composed into high-level business processes. People wishing to prepare their summer vacation can for instance, access a specialized web portal and specify their needs and expectations in terms of just to cite a few favorite airline carrier, accommodation type, and quality of response to be returned. Summer vacation scenario requires the collaboration of at least four E-services: flight reservation, hotel booking, attraction searching, and user notification. All these E-services have to be inter-connected according to a certain flow of control. For instance, flight reservation is completed first, before both hotel booking and attraction searching are initiated. Moreover, various constraints may influence the control flow of the composition and execution of E-services. For instance, constraints on expenses and overlapping dates have to be considered in the flow definition.

With the latest developments in wireless technologies, new Web services are put forward for the benefit of persons who are most of the time on the move. These persons heavily rely on mobile devices to conduct their operations. M-services denote these Web services and are meant to be either remotely triggered from mobile devices for their execution or transferred from provider sites to the mobile devices of users on which their execution occurs [3].

It is largely accepted that composing services, rather than accessing a single service, is essential. Searching for the relevant services, composing them into a composite service, triggering them, and monitoring their execution are among the operations that users will be in charge. Most of these operations are complex, although repetitive with a large segment suitable to computer aids and automation. Therefore, SAs are appropriate candidates to assist users in their operations. Two types of SAs are considered. User-agents acting on behalf of users. And, provider-agents acting on behalf of providers thus, offering Web services (E-services or M-services) to users.

The integration of SAs and Web services into the same environment raises the importance of a specification approach. We advocate to deploy the specification at three levels: intrinsic, organizational/functional, and behavior. Each level has multiple properties that vary according to the component, either agent or service, to which the level is applied.

[1] In the following, Web service and service are used interchangeably.

Web services

An E-service is a component that an organization provides in order to be assembled and re-used in a distributed, Internet-based environment. An E-service presents the following features [1]: independent as much as possible from specific platforms and computing paradigms; developed mainly for inter-organizational situations rather than for intra-organizational situations; and easily composable.

The weak definition of an M-service is to remotely trigger an E-service from a mobile device for execution. The strong definition is to consider an M-service as a specific type of E-services whose execution takes place on top of a mobile device. In that case, the M-service has to meet the following requirements [3]: transportable through wireless networks; composable with other M-services; adaptable according to the computing features of mobile devices; and runnable on mobile devices.

Service chart diagrams

Represented in Fig. 1, a service chart diagram is used for the specification of a service [2]. This diagram is an extension to the traditional state chart diagrams of UML.

In addition to representing the states that a service takes, a service chart diagram wraps these states into four perspectives, each perspective having a set of attributes (Fig. 1). The *flow* perspective corresponds to the chronology of executing a set of connected services (previous services/next services attributes). The *business* perspective identifies the organizations that are ready to provide a service (business

Proceedings of the Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'03)
1080-1383/03 $17.00 © 2003 IEEE

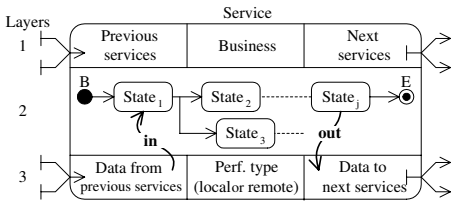

Figure 1. Service chart diagram

attribute). The *information* perspective identifies the data that are exchanged between services (data from previous services/data for next services attributes). Finally, the *performance* perspective illustrates the way a service is invoked for execution whether remotely or locally (performance type attribute).

The specification approach

The specification approach that we suggest has three levels: intrinsic, organizational/functional, and behavior. The properties of each level are below presented.

a. Software agents

Intrinsic level consists of *identifier*, *role*, and *type* properties.

Organizational level consists of *domain visited* and *domain not-visited* properties.

Behavior level consists of *state diagram* property.

b. Web services

Intrinsic level consists of *identifier*, *description*, *type*, *input arguments*, *output arguments*, *execution cost*, and *execution time* properties.

Functional level consists of *component link*, *mandatory causal link*, and *optional causal link* properties.

Behavior level consists of *state diagram* property that is wrapped into a service chart diagram.

Agentification of an environment of Web services

For the agentification needs of an environment of Web services, we decided to deploy an *agent-based multi-domain* architecture (Fig. 2). Domains are spread across the network and administrators maintain them. Two types of domain exist: *user-domain* and *provider-domain*. We assumed the existence of one user-domain and several provider-domains. A domain is a computing platform on top of which portal of services and agents are deployed. Users browse the portal of C-services from different devices: fixed devices such as desktops and mobile devices such as personal digital assistants. More details on the operating of the agent-based multi-domain architecture are given in [4].

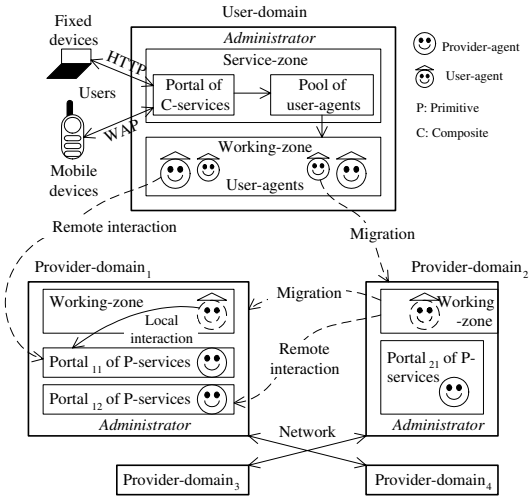

Figure 2. Multi-domain architecture

Summary

We presented an overview of an approach that uses software agents in the specification of a Web service-oriented environment. The approach suggests three levels: intrinsic, organizational/functional, and behavior. The three levels are applied to agents and services in a different way. During the agentification of services, two types of software agents have been suggested: user-agent and provider-agent. User-agents are associated with user-domains. And, provider-agents are associated with provider-agents.

References

[1] B. Benatallah, Q. Z. Sheng, and M. Dumas. The Self-Serv Environment for Web Services Composition. *IEEE Internet Computing*, 7(1), January/February 2003.

[2] Z. Maamar, B. Benatallah, and W. Mansoor. Service Chart Diagrams - Description & Application. In *Proceedings of The Twelfth International World Wide Web Conference (WWW'2003)*, Budapest, Hungary, 2003.

[3] Z. Maamar, W. Mansoor, and Q. H. Mahmoud. Software Agents to Support Mobile Services. In *Proceedings of the 1st International Joint Conference on Autonomous Agents & Multi-Agent Systems (AAMAS'2002)*, Bologna, Italy, 2002.

[4] Z. Maamar, Q. Z. Sheng, and B. Benatallah. Selection of Web Services for Composition Using Location of Provider Hosts Criterion. In *Proceedings of The Ubiquitous Mobile Information and Collaboration Systems Workshop (UMICS'2003) held in conjucntion with The 15th Conference On Advanced Information Systems Engineering (CAiSE'2003)*, Klagenfurt/Velden, Austria, 2003.