On Promoting Ad-Hoc Collaboration Among Messengers

Zakaria Maamar
Zayed University
Dubai, U.A.E
zakaria.maamar@zu.ac.ae

Qusay H. Mahmoud
University of Guelph
Guelph, Canada
qmahmoud@uoguelph.ca

Abdelouahid Derhab
CERIST
Algiers, Algeria
aderhab@mail.cerist.dz

Abstract

The explosion growth in the market place for handheld wireless devices has enabled new opportunities for wireless applications. Currently, handheld devices are restricted to being clients that make requests to servers and receive responses over the network. But as mobile ad-hoc networks become the trend, such devices will need to become active participants that serve requests from other devices and convey data to other devices as well. In this paper we present our vision of the future role that handheld devices will play in a mobile ad-hoc network configuration. We present this vision as part of the MESSENGER project that develops data management mechanisms for UDDI registries of Web services using mobile users and their software agents, and then describe its extension for exchanging descriptions of Web services during ad-hoc collaboration sessions. User agents are in charge of interacting with peer users over an ad-hoc network, and collaborating on feeding UDDI registries with recent content.

Keywords. Web service, UDDI registry, Ad-hoc.

1. Introduction

In [6], we reported on the MESSENGER project that aims at developing mechanisms to support data management among a set of distributed UDDI registries. Unlike other Web services initiatives, this project’s concerns are as follows: several UDDI registries are deployed, there is no pre-defined communication infrastructure between the UDDI registries (but a wireless infrastructure was privileged), and absence of a central component to coordinate the UDDI registries. The MESSENGER project suggests combining users and software agents [1] into what we refer to as messengers. Initially, software agents reside in users’ mobile devices (i.e., handheld) and cache a description of the Web services that permitted satisfying the needs of these users. Each time a user is in the vicinity of an UDDI registry, her software agent interacts with that UDDI registry so details stored in her mobile device on Web services are submitted. In this paper data and details are interchangeably used, and both refer to the description of a Web service.

In a typical Web services scenario, an UDDI registry is involved in two operations. The first operation is to receive the announcements of Web services (also called services in this paper) from providers. And the second operation is to search its content for the services that satisfy users’ needs. However, since the providers’ announcements of services are submitted to separate UDDI registries, this leads to a different content among the UDDI registries. Developing mechanisms to support the exchange of content among independent UDDI registries is deemed appropriate.

In the current operation mode of messengers [6], descriptions of Web services are first, collected from UDDI registries and later on, distributed to other UDDI registries subject to satisfying various criteria (Section 2.2). This form of distributing Web services does not foster the tremendous capabilities that offer both wireless technologies and mobile devices [3]. For instance if mobile devices are in the vicinity of each other (i.e., reachable) they can on the spot form a mobile ad-hoc network, which allows the exchange of data between these devices. In such a network devices serve as routers for others, which permits data to flow from one device to another according to a specific routing protocol [4].

In this paper we discuss how ad-hoc collaboration among messengers occurs and how it is integrated into the operation of messengers. By authorizing messengers to engage their peers in collaborative situations, the exchange of data among UDDI registries is boosted. This not only allows collecting descriptions of Web services from UDDI registries as is currently happening [6], but also allows collecting additional descriptions of Web services from messengers. This way of doing has several advantages, but at the same time poses various challenges and arises various issues, which simply highlight the complexity introduced by the dynamic and volatile natures of ad-hoc networks. The rest of the paper is organized as follows. We overview next the MESSENGER project. Then we describe the ad-hoc collaboration among messengers. Finally we discuss some
simulation results and draw our conclusions.

2. The MESSENGER project

2.1. Background

In the MESSENGER project each UDDI registry is related to a structure known as cluster of Web services. Several clusters are deployed across the communication infrastructure, so providers can connect to the most appropriate cluster based on various criteria like proximity, bandwidth, and workload. The connection between providers and clusters is of type wired (could be too of type wireless). For tracking purposes a provider only connects to a single cluster. Thus it cannot announce its services in multiple UDDI registries. The cluster on which a provider posts its Web services for the first time is known as master. Interesting are the situations where providers have similar Web services but respectively announce them in distinct UDDI registries. Unless some appropriate mechanisms are developed, an UDDI registry would never be aware of the existence of similar services in other UDDI registries. Furthermore, for a user wishing to satisfy her needs by triggering or composing services [2], she should be able to use all the existing services regardless of where they are advertised.

While residing in the user’s mobile device, its software agent caches a description of the Web services that permitted satisfying the needs of this user. Later on and acting on behalf of providers, users announce Web services in various UDDI registries to be related to clusters of type slave. Since users have mobile devices, mobile support stations manage these devices by identifying their physical location, handling their incoming/outgoing messages/calls, etc. A mobile support station interacts with mobile users within its radio coverage area known as wireless cell. When a user enters a new cell (i.e., she is now under the coverage of a new mobile support station), an exchange of details happens between the agent of the user and the UDDI registry. This exchange consists of updating the UDDI registry content according to various scenarios described in [6]. It is important to highlight that a user does not have to interact with all the clusters. Her association with a mobile support station (i.e., an UDDI registry) depends on her routes to places like mall, school, etc.

Because a UDDI registry receives descriptions of Web services from two sources, providers of Web services and agents of users, the services are classified as internal and external. Internal services are announced in an UDDI registry of a master cluster (providers take care of the announcements). This UDDI registry has a full control over the internal services by guaranteeing for example their QoS. External services are always announced in an UDDI registry of a slave cluster (agents take care of the announcements). This UDDI registry cannot guarantee for example the QoS of the external services and their availability in their respective provider hosts during invocation.

2.2. Messengers in operation

The agentification of the data management of UDDI registries necessitated three types of software agents (Fig. 1): provider-agent, UDDI-agent, and user-agent. Usually, the coverage areas (represented by circles in Fig. 1) of the mobile support stations overlap. However to keep the figure clear the overlapping is not shown. Since the clusters of Web services are wirelessly connected, a continuous and reliable exchange of the content of the different UDDI registries cannot be guaranteed. Fig. 1 also depicts the notion of messenger: (user, user-agent) on the move. Users with their mobile devices are always related to a particular mobile support station. Whenever a user moves to a new place, which is outside the coverage area of a mobile support station, a handover occurs between this station and the new mobile support station that manages this new place.

A provider-agent identifies a provider (i.e., a business) that posts its Web services on the UDDI registries of the multiple clusters. However, a provider is only authorized to connect to one cluster of type master. The services announced in the UDDI registry of a master cluster are labelled as internal. This labelling helps (i) identify the UDDI registry where the services are announced for the first time, (ii) know the identity of the providers to whom the services belong in case their identity can be revealed, (iii) deny attempts of user-agents to update the description of the services, and (vi) name the UDDI-agent that ensures that the execution parameters of the services (e.g., execution cost, availability rate) are satisfied during execution.

Since a provider only announces its Web services in one UDDI registry, messengers take care of the UDDI registries of the remaining clusters of type slave. For announcement purposes in slave clusters, the messengers ought to bind to various policies [6]. The services posted on the UDDI registry of a slave cluster are labelled as external. This labelling helps (i) indicate that a third entity (i.e., a certain messenger) has announced the services, (ii) inform that the services can always be subject to unpredicted changes at the level of their respective providers, and (iii) state that the UDDI-agent cannot guarantee the execution parameters (e.g., response time) of the services.

Because users are engaged in announcing external Web services, the agreement of the providers of these Web services is required. For different reasons such as privacy (e.g., a provider does not want to announce all its services in a certain UDDI registry) and trustworthiness (e.g., a provider is not confident in the security mechanisms of a certain UDDI registry), a provider clearly states in the de-
scription of a Web service, which is already submitted to the UDDI registry of a master cluster, whether this service can be announced in the UDDI registries of slave clusters. The statements of a provider are done through policies, whose execution outcomes are appended to the announcements of services. The statements are intended to the messengers that roam the different clusters. It should be noted that all the services of a provider have to be posted on at least one cluster of type master. Fig. 2 is an example of a provider-defined trustworthiness policy.

A user-agent resides in the mobile device (e.g., cellphone) of a user constituting both a messenger when they move. The main duty of a user-agent is to satisfy its user’s needs (e.g., car rental) after identifying and selecting the relevant Web services with the help of an UDDI registry. To this purpose, the user-agent initiates interactions with the UDDI-agent of an UDDI registry. The selection of a particular UDDI registry depends on the current location of the user with regard to the mobile support station that manages her mobile device. Once the services are identified and triggered for execution, the user-agent caches in the user’s mobile device various details like the identifier of the services, the UDDI registry with whom the user-agent has dealt with in order to obtain the services, and the providers of services and their location according to the master clusters. A refinement of the details that user-agents store is given in Section 3.2. These details can be potentially announced to distinct UDDI registries of slave clusters for further processing. This happens after verifying the authorization policies of the announcements (Fig. 2).

2.3. Prototype

A prototype of the MESSANGER approach for the data management of UDDI registries was developed [6]. The prototype uses Sun’s Java WSDP 1.5, which is an integrated toolkit for building, testing, and deploying Web Services. Java WSDP comes with an implementation of an UDDI registry, which we integrated into the prototype. For the client side, we used Sun’s J2ME Wireless Toolkit, which provides an implementation of J2ME. The setup of the prototype environment was as follows: (i) three 802.11b wireless LANs were installed to cover an area of 100 meters by 300 meters; (ii) one UDDI registry was installed within the range of each wireless LAN; and (iii) users were equipped with PDAs having each a wireless access card to connect to the LANs and MIDP4Palm implementation installed.

For illustration purposes our running scenario depicted a user who wants to find a print service that identifies the closest printer from her current location. Once the print service is identified, the user-agent needs to seek the authorization of the user prior to posting this service on other UDDI registries. The provider of the print service has to authorize this posting, too.

3. Ad-hoc collaboration among messengers

An ad-hoc network is a local area network or other small network, especially one with wireless or temporary plug-in connections, in which some of the devices (sometimes mobile) take part in the network only for the duration of a communication session, or because these devices are in close proximity so a communication session can occur.

According to Ishibashi and Boutaba [5], mobile devices in a mobile ad-hoc network have a different role than in a conventional local-area network. In this latter network, communications are centered around base stations. The infrastructure up to a base station is mostly fixed, so the topology is stable. Some elements of Fig. 1 like providers and UDDI registries illustrate a local-area network and have, to a certain extent, framed the first development steps of the MESSANGER project. In an ad-hoc network, mobile devices act not only as end-systems but also as routing devices. In this part of the paper, we aim at looking into the value-added of motivating messengers to engage in ad-hoc collaboration. We first, discuss the rationale of this collab-
if trustworthiness: (provider_agent, uddi_registry) < 0.5
then user_agent: not(announce(web_service, in(uddi_registry)))

Figure 2. Example of a provider-defined policy

Rule description. provider-agent, forbids to any user-agent the announcement of its Web services in UDDI-registry, if the trustworthiness value between this provider and this UDDI registry is less than 0.5.

oration and the challenges it faces and second, present the appropriate mechanisms we envision to perform this collaboration.

3.1. Motives and challenges

The mutual awareness of distant messengers supports the emergence of an ad-hoc network upon which mechanisms for conducting collaboration can operate. Prior to this collaboration, it is assumed first, that messengers have the necessary collaboration authorizations of their respective users and second, that messengers are in the vicinity of each other (vicinity depending on the technical characteristics of the wireless cards). Currently, routes of users and announcement policies limit the messengers in their exchange of service descriptions with UDDI registries [6]. This exchange can be boosted by allowing messengers to collect additional descriptions from their peers. It is expected that this description collection does not violate the multiple policies, which frame the operation of messengers (Section 2.2).

In Fig. 1, the current scenario for data management among distributed UDDI registries highlights a messenger that conveys data on a single user. The scenario that is targeted through the ad-hoc collaboration is to enable messengers to convey data on multiple users. In Fig. 3, messengers1,2,3 are in close proximity and agree on establishing an ad-hoc network. After exchanging data to be detailed in Section 3.2, each messenger possesses now extra service descriptions from the other two messengers. For example messenger3 can now post additional service descriptions on an UDDI registry. These descriptions are split into two: those associated by default with user3 and those associated with users1,2. The selection of messenger3 among the three messengers to submit descriptions happens according to the designated-messenger principle, which we discuss in Section 3.2. In Fig. 3, links between users’ mobile devices are formed within direct communication range, and devices and links both combine to create the network topology. During the lifetime of an ad-hoc network, messengers may move around and within the network, altering the topology by setting or breaking links between devices.

Multiple aspects need to be investigated when messengers are engaged in ad-hoc collaboration including:

What are the mechanisms that need to embed users’ mobile devices, so messengers can detect the opportunity of collaborating independently of the opportunity of detecting their proximity?

What are the steps that messengers perform once they agree on collaborating, what are the steps to conduct in order to join/leave an existing ad-hoc collaboration of messengers without disrupting the data exchange, and what are the incentives to join such a session and sharing descriptions of Web services with peers?

Is there a need to distinguish the description of a Web service that is directly submitted to UDDI registries vs. the description of a Web service that is indirectly submitted to UDDI registries through multiple messengers?

How to control/limit the number of messengers in an ad-hoc collaboration session, what are the reasons of this control, and how to avoid the overloading of messengers according to their computing and storage capacities?

While some of the aforementioned aspects are related to ad-hoc networking technology [3], others are directly related to the strategy of distributing descriptions of Web services among UDDI registries that the MESSENGER approach has defined [6]. For instance, detecting the vicinity of two mobile devices is a technical detail that depends on the protocols that manage the equipments of these devices. However the steps that messengers perform prior to collaborating are subject to some policy verification.

3.2. Scenarios during ad-hoc collaboration

In [6], we detailed the nature of the description of a Web service. This description is stored in a user’s mobile device and has the following content: service name (i.e., identifier), UDDI-registry identifier with regard to the master cluster wherein the service is announced as internal (needed for
invocation purposes), outcomes of announcement policies (e.g., list of UDDI registries where a service cannot be announced), and extra details for service selection purposes. The description of a Web service that a user-agent caches in a mobile device is specified in XHTML.

The ad-hoc collaboration among messengers sheds the light on four scenarios that are separately approached because of the challenges each poses and the operation mechanisms each requires. Indeed it is reported that scalability, efficiency, and resilience to unannounced disconnections and frequent network topology change are key concerns in ad-hoc networking [4]. In the following we assume that messenger_i and messenger_j are engaged in ad-hoc collaboration.

**Scenario 1.** Both messenger_i and messenger_j have different descriptions of Web services, respectively. The ad-hoc collaboration has a positive impact on both. These messengers can now collect details on new Web services. Each messenger reports the source from which it obtained the additional descriptions of Web services. Enriched with these descriptions, the messengers terminate the communication session and resume announcing these descriptions to other UDDI registries or exchanging these descriptions with other peers.

**Scenario 2.** Messenger_i possesses descriptions of Web services that are not available at the level of messenger_j. The ad-hoc collaboration has a positive impact on messenger_j. This messenger can now collect details on new services, which were not included in its initial list of services. In the description of any extra Web service, messenger_j indicates messenger_i as the source of this description. Enriched with the descriptions of the new Web services, messenger_i terminates its communication session with messenger_i and starts announcing this description to other UDDI registries or exchanging these descriptions with other peers.

**Scenario 3.** Both messenger_i and messenger_j have similar descriptions of the same Web services. It is suggested that these messengers first, refrain from continuing their communication and second, cancel the ad-hoc collaboration so resource consumption is avoided. This collaboration does not contribute to enhancing the descriptions of the Web services that both messengers possess.

**Scenario 4.** Both messenger_i and messenger_j possess contradictory descriptions of the same Web services. This constitutes a challenge since conflicting details need to be sorted out so these descriptions become consistent among messengers. An example of inconsistency is the execution cost of a Web service. Although a Web service originates from the same provider, a different execution cost is given in each description. One of the reasons of this difference could be the date on which a messenger has collected the description of this Web service from either an UDDI registry or another messenger. As a solution the description that is considered as recent, is taken into consideration, which permits to one of the messengers to update its description of the Web service. Once the conflicts are resolved, additional operations occur as per Scenario 3.

The opportunity of collaboration between messengers is motivated by three elements. First, the messengers expose a cooperative attitude through their willingness of sharing descriptions of Web services. Second, users can benefit from extra descriptions that could help satisfy their needs. And third, providers could reward the messengers that helped disseminate information on their Web services. Assessing rewards is possible because of the invocation requests that providers receive from users to their Web services. We recall that these requests are used for calculating the trustworthiness level of a provider (Fig. 2).

Before an ad-hoc collaboration session is over, the participating messengers have to agree on the messenger that will be designated for feeding an UDDI registry with descriptions of Web services. This assumes that the messengers are all related to the same mobile support station, i.e., to the same UDDI registry. The rationale of designating a messenger is to avoid the transfer of duplicate descriptions from the participating messengers in an ad-hoc collaboration session to an UDDI registry. This messenger could be selected using various criteria like storage capacity and highest connectivity. It should noted that a user can still set the maximum size of the descriptions of Web services she would accept storing in her mobile devices.

### 4. Simulation results

In this section, we present some simulation results about the performance of the MESSENGER approach using GloMoSim (Global Mobile Simulator) [7]. The simulation environment considers an area of 1500m × 1500m and encompasses 4 mobile support stations and 16 mobile users. Each mobile user randomly selects a cell to whom it moves with a random speed uniformly distributed between 0 and a certain maximum speed $V_{max}$. Then, the user suspends moving for about 30 seconds before resuming his movement to a new cell. The radio coverage area is set to 250m. Each simulation runs for 60 minutes. Initially, 10 descriptions of Web services are posted on each UDDI registry. For performance purposes, we define two metrics:

- **Description availability:** is the ratio of the number of Web services descriptions stored in an UDDI registry over the total number of descriptions posted on all UDDI registries.
- **Message overhead:** is the total number of messages exchanged during the simulation.
In Fig. 4, we notice that as time progresses, the movement of messengers towards different cells increases the availability of descriptions. Moreover, we notice that permitting ad-hoc collaboration among messengers gives higher description availability than the original operation mode of the MESSENGER approach.

5. Conclusion

We reported on our experience with the MESSENGER project that aims at developing data management mechanisms among a set of distributed UDDI registries in a wireless environment. In addition we outlined an agent-based approach for promoting ad-hoc collaborations among a group of peers who are mobile users. We provided four operating scenarios that address the challenges in mobile ad-hoc networks including scalability, efficiency, and resilience to unannounced disconnections and frequent network topology change. Our vision of the future role of users is backed by the latest development of handheld devices. Such devices continue to become more capable and thus will be able to serve requests to other clients instead of just requesting services.

References