Software Agents and Wireless E-Commerce

Zakaria Maamar[1,2], Hamdi Yahyaoui[3], Wathiq Mansoor[1], and Willem-Jan vd Heuvel[4]
[1]Zayed University, United Arab Emirates.
[2]Institute for Technological Innovation, United Arab Emirates.
[3]Laval University, Canada.
[4]Tilburg University, The Netherlands.

Software agent technology will become a necessity to e-commerce (traditional or wireless) rather than luxury. We propose an agent-based environment, which we call E-Commerce through Wireless Devices (E-CWE), that allows users, stationary or mobile, to submit their requests for services offered by providers. The E-CWE environment consists of a reception platform including a repository of service descriptions. A supervisor-agent is in charge of this repository. Acting on behalf of their users, user-agents are dynamically created in this platform. In the E-CWE environment, security is achieved in two steps: securing service access and securing payment. A prototype of a Travel Planning Agent (TPA) system is proposed to evaluate the performance of the environment.

Additional Key Words and Phrases: Software agents, e-commerce, and wireless devices.

1. OVERVIEW

Imagine the following scenario: it is freezing outside and you are at home, browsing various on-line catalogues of summer-sport products. With the recent progress of information technologies, mainly the Internet, this situation is becoming ubiquitous. Unfortunately, on-line shopping usually known as e-commerce [IBM 2001], is not as transparent as we would like. In fact, novice users could face different obstacles. First, users need to be able to search for the relevant web sites that provide on-line access to catalogues. Second, they need to understand how these web sites work. Third, they need to specify their needs according to these sites' characteristics, e.g. terminology. Fourth, in order to compare offerings, they need to retain information, such as prices, warranties, and returning policies, while switching from one site to another. Last but not least, they need to be aware of security problems when submitting sensitive information, such as their credit card numbers.

In the above-mentioned hypothetical situation, we assumed that this person has a workstation that could be constantly connected to a high bandwidth Internet-

Address: contact author - Systems Engineering Group@ZU, College of Information Systems, Zayed University, PO Box 19282, Dubai, U.A.E. zakaria.maamar@zu.ac.ae.

connection. Unfortunately, this assumption is not always valid. It may happen that this person would like to buy a gift for his son while he is returning home from his office. Using his mobile phone, our person accesses several on-line stores looking for the perfect gift. His search could be narrowed by the maximum price he is willing to pay, the delivery time, and the age of his son. All these operations have to be monitored from the display of this person's mobile phone. This is really a complex scenario. Therefore, it becomes urgent to support persons, stationary or mobile, in this kind of activity. This is our aim in the E-CWE (E-Commerce through Wireless Devices) project.

The highlights of the E-CWE environment are as follows: associate users with user-agents; embody user-agents with personalization and mobility mechanisms; and associate providers with provider-agents.

The remainder of this paper is organized as follows. Section 2 introduces the E-CWE environment in terms of architecture and operating mode. Section 3 discusses security aspects in this environment. Section 4 examines an implementation of the E-CWE environment. Section 5 consists of concluding remarks.

2. THE E-CWE ENVIRONMENT

2.1 Architecture

Figure 1 represents the architecture of the E-CWE environment. This architecture consists of a network of different providers that have decided to integrate their way of conducting business, for instance by offering on-line access to their services[1]. We assume that providers are associated with different types of sites that could range from simple ones such as catalogues to complex ones such as auction markets. In the E-CWE environment, users entrust their routine and complex operations to user-agents. In Figure 1, a repository of services exists. Providers subscribe to this repository. Details on the repository of services will be given later. In the same figure, dashed lines represent wireless interactions. Such interactions should be minimized in case charges on connections apply. Connections should also be secure (cf. Section 3).

The repository of services is the backbone of the E-CWE environment and consists of two types of services: user-services and provider-services. A supervisor-agent is in charge of this repository. User-services are offered to users and give an overview of what kinds of needs could be satisfied. Examples of user-services could be stock quote monitoring and travel planning. In order to be satisfied, user-services require provider-services that correspond to providers' capabilities in the repository of services. After the supervisor-agent has granted access to providers, they advertise their capabilities in this repository (cf. Figure 1). In the rest of this paper, I-services will denote provider-services. "I" stands for Intelligent[2].

[1] This implies that providers have to commit their resources to the success of the services that will be offered by the E-CWE environment.

[2] The word intelligent is used in the sense that services are able to negotiate on behalf of providers they are associated with.

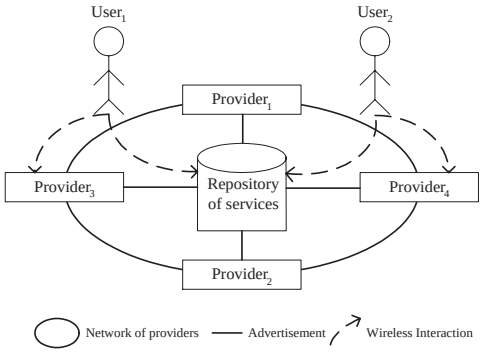

Fig. 1. Proposed architecture of the E-CWE environment

2.2 Operating Mode

In the E-CWE environment, we assume that users must subscribe to this environment, i.e. membership is required and prohibits the access to unknown users. Initially, each user must download, from the repository of services to his mobile device, the list of user-services that exist and save them for later use. Then, the user would be able to display this list and browse its content. Transferring the list of user-services to users' devices avoids frequent connections to the repository of services and hence, to networks. Each time the list of user-services is updated, e.g. a user-service is added; this list is modified and then submitted to the E-CWE members through broadcasting. In Figure 2, operation (0) illustrates the membership assumption.

Figure 2 represents the operating mode of the E-CWE environment. In what follows, numbers between brackets correspond to the chronology of operations. Initially, the user browses the previously downloaded list of available user-services on his mobile device (1). According to his needs, the user selects the appropriate user-service. Next, he specifies his needs and sends them to the E-CWE runtime environment[3], specifically to the supervisor-agent of the repository of services (2). This is the first time that a wireless connection occurs in the E-CWE environment. At this stage, the user disconnects from the network and waits for the results to be delivered. Users' needs could be classified into two types [Marathe and Diwakar 2001]: must-have needs and would-like-to-have needs. The second type of needs could be used for relaxing certain constraints, in case, for example, the search for relevant providers fails or the negotiation with providers fails. In a travel-planning scenario, examples of must-have needs could be the budget limit and the number of people. Examples of would-like-to-have needs could be the stay duration and the type of transportation. As soon as the supervisor-agent receives the user's needs, it assigns a user-agent to satisfy them.

The next step for this user-agent is to be transmitted to the reception platform on

[3]In [Mahmoud 2001], the author argues that mobile devices have limited resources, and thus it would be virtually impossible to run an agent platform on a cellular phone, for example. To overcome this limitation, the author runs his agent platform on a remote host denoted by agent gateway. This platform is similar to the E-CWE runtime environment we suggest.

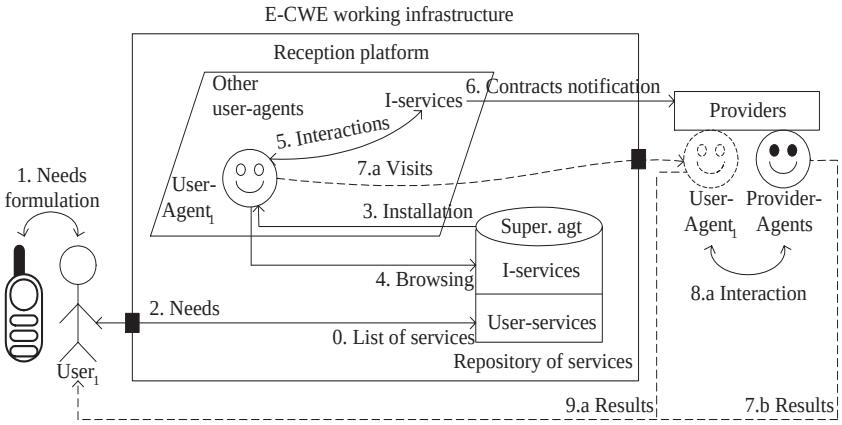

Fig. 2. Proposed operating mode of the E-CWE environment

which its installation starts (3). This platform is mainly a runtime environment on which agents could undertake their operations. Once the user-agent is within this platform, it starts browsing the repository of services looking for the I-services that are required to satisfy its user's needs (4). In fact, users' needs are matched with I-services [vd Heuvel 2001]. When relevant I-services are identified, the user-agent interacts with them in the reception platform (5). Interactions between user-agents and I-services could lead into negotiations that end up with contracts signature. Contracts document the agreed upon obligations and commitments of both users and providers. For security reasons, providers as well as their provider-agents are notified about the contracts the I-services have signed on their behalf (6). I-services are in charge of this notification. After signed contracts, two situations take place in the E-CWE environment. Letters **a** and **b** of Figure 2 denote both situations.

—Situation **a** called yes-visit: here, the user-agent has to visit the provider sites in order to get the information that satisfies its user's needs. In fact, the interactions that this agent has had with I-services are authorizations to use the resources of these providers. The yes-visit case fits well the environments in which the user is involved in operations, e.g. auctions.
For the user-agent, contracts permit to design the itinerary of its visits to provider sites. When the itinerary is set up, the user-agent visits the different providers (7.a), interacting with their provider-agents (8.a). Finally, the user-agent sends results to the user (9.a).

—Situation **b** called no-visit: here, the user-agent does not have to visit the provider sites. Instead, the I-services inform the provider-agents about the actions to be carried out in order to satisfy users' needs, e.g. goods to be shipped, reports on funds to be emailed, etc. Travel planning belongs to the no-visit case. In real life, travel agencies are responsible for getting tickets on behalf of their customers and sending these tickets to them.
After receiving notifications from their I-services about the contracts they signed, provider-agents start implementing these contracts. First, provider-agents check

their validity. Then, they send results to the user (7.b).

—Remark: a third situation is possible and combines both situations **a** and **b**. We do not discuss it (we expect that it would be complex).

3. SECURITY IN THE E-CWE ENVIRONMENT

Authentication and confidentiality are among the E-CWE environment's requirements in security. Authentication is about checking the identity of participants. Confidentiality is about exchanging sensitive messages such as user credit card number when payments are due.

In authentication, providers should be able to check user-agents' identity before they allow them to enter their sites. This identity check should be preceded by contract authentication (cf. Figure 2, operation 6). Providers could also be the subject of authentication by users. In confidentiality, user-agents' payments to the services they receive from providers should be done in a secure way.

We consider different technologies for meeting both security requirements. SPKI (Simple Public Key Infrastructure) is the one that interests us [Ellison et al. 2001]. SPKI relies on publishing public-keys to potential participants. In addition, SPKI defines two elements: 1) certificate formats and semantics and 2) the process to verify the validity of certificates. In SPKI, a pair of keys (private and public) is associated with a participant (provider and user). The private key signs certificates. Signed by the private key of the provider, a certificate gives to the user the rights to access a service for a limited period. Each user wishing to request a provider's service must have a certificate to access this service. Regarding the confidentiality requirement, cryptographic protocols could satisfy this requirement.

In the E-CWE environment, security in achieved in two steps: secure service access and secure payment.

(1) The purpose of the first step is to secure the access to services. Providers (here, we refer to provider-agents) authenticate users (here, we refer to user-agents) by exchanging certificates. We recall that unknown users are not accepted in the E-CWE environment, according to the membership principles. A provider signs a certificate using its private key. Then, it sends this certificate to the user. This user verifies the certificate that he receives using the provider's public key. Then, the user signs this certificate using his secret key. Finally, the user returns the certificate to the provider who will verify it using the user's public key.

(2) In the second step, securing communication of payment operations should occur between the provider and the payment system as well as between the user and this payment system. The first operation is an authentication between the payment system and the provider. Afterwards, the provider sends to the payment system its account number, the services' costs and the certificates it obtained in the "secure service access" step. The public key of the payment system is used to sign each of these messages. Next, the payment system sends a message to the user that contains the service cost and a certificate signed by its secret key. The user decrypts this message by the public key of the payment system and verifies the service cost. Then, he adds to this message his credit number card and signs it with his secret key then by the public key

of the payment system. This prevents a third party from reading the message. Afterwards, the payment system uses its secret key in order to decrypt the messages it received from the provider and the user as well. To this end, it relies respectively on this provider's public key and this user's public key, as well. Finally, the payment system completes all the process. If needed, the payment system could communicate with other payment systems.

4. IMPLEMENTATION OF THE E-CWE ENVIRONMENT

In the E-CWE environment, user-agents carry out operations on behalf of users. Such agents may have to move from the E-CWE runtime infrastructure to users' mobile-devices. In addition, these agents may have to move from this infrastructure to provider sites. Before any move happens, the agents have to look for the providers on the basis of users' needs. Several programming languages to implement software agents exist. Those that we are interested in should adhere to the following criteria: allow mobility, be platform independent, and offer look-up services. Java combined with Jini seems to meet these features[4]. It is accepted that Java is suitable for the construction of agent-based systems. Regarding Jini [Sun 2001], it is intended to deal with open environments' issues, by providing, among other things, a dynamic method for locating services on a network. In the E-CWE environment, Jini will be applied to implement the repository of services and supporting facilities, such as searching for providers' services, matching needs with services, notifying when services of interest become available, etc. Negotiation support between users and providers should be dealt with separately.

In order to illustrate how the E-CWE environment is useful to conduct wireless e-commerce, we are currently prototyping a Travel Planning Agent (TPA) system. The TPA scenario belongs to the no-visit case (cf. Figure 2, case **b**). Travel planning is a non-trivial application; different aspects have to be considered including which airlines to take, in which hotels to stay, and what places to visit. The chronology of operations in TPA is presented below:

—Associate user's needs with blocks; each block will contribute to satisfy a part of these needs. In TPA, examples of blocks could be trip itinerary, hotel booking, and sightseeing. Blocks are refined according to must-have needs and would-like-to-have needs.

—Determine the sequence of navigation between the different blocks. The objective is to define temporal and precedence relationships between blocks if such relationships do exit. For instance, a hotel booking block will depend on the trip itinerary block. Meanwhile, hotel booking and sightseeing blocks could be conducted simultaneously.

—Refine the blocks into primitive actions that implement them. For instance, the trip itinerary block includes locating airlines on the requested date, determining of seats availability, requesting quotation from other airlines, making decisions regarding the purchase, and finally making payments. The making decision action could require the user's approval, before it is implemented.

[4]The Java 2 Micro Edition (J2ME, java.sun.com/j2me) meets the requirements of wireless systems.

—Finally, look for the services that satisfy these actions. To this end, Jini lookup service is involved. The performance of this current step has prerequisites that should be met first:
  —The providers register themselves at the level of Jini lookup service. In fact, the providers will receive a "registrar" object from Jini that will identify them.
  —Through the "registrar" object, providers advertise their services. Each service has attributes that describe it. Then, user-agents send requests containing the values of the desired attributes. A matching process occurs between the values received from the client and the attributes describing the service.

5. CONCLUSION

In this paper, we presented the E-CWE environment. It aims at achieving wireless secure e-commerce operations. This environment has several features:

—Users do not need to keep connecting to the Internet waiting replies for their requests. This is achieved by creating mobile user-agents to deal with these requests. Users are notified when replies are ready.

—Providers do not need to keep connecting to the Internet waiting for customers for their services. This is achieved by creating I-services that act on behalf of providers. These services could contact providers for consultation or approvals of contracts.

Regarding security in the E-CWE environment, SPKI ensures authentication and confidentiality. However, a trust management mechanism [Singh and Venkatraman 2001] is needed to evaluate the quality of services provided by the providers and even the evaluation of the seriousness of users. This trust management is scheduled for future modifications of the E-CWE environment.

Additional issues are also considered. Among them, the repository of services may be inflated with unnecessary information. And, the reception platform may become a bottleneck. The E-CWE performance is also considered. Several elements should be assessed, for instance the number of keystrokes, the number of user wait seconds, the number of server wait seconds, the number of re-connects required, etc.

ACKNOWLEDGMENTS

The authors would like to thank the reviewers for their useful comments.

REFERENCES

Ellison, C., Frantz, B., Lampson, B., Rivest, R., Thomas, B., and Ylonen, T. Visited Jan. 2001. Simple public key certificate. IETF RFC 2693, http://world.std.com/ cme/spki.txt.

IBM. Visited Feb. 2001. Ibm institute for advanced commerce. http://www.ibm.com/iac/.

Mahmoud, Q. 2001. Mobiagent: a mobile agent-based approach to wireless information systems. In *Proc. of Agent-Oriented Information Systems workshop at Autonomous Agents'20001 conference* (Montreal, Canada, 2001).

Marathe, M. and Diwakar, H. Visited Feb. 2001. The architecture of a one-stop web-window shop. Official newsletter for the ACM: Special Interest Group on Electronic Commerce (SIGEcom). Winter issue, http://www.acm.org/sigs/sigecom/exchanges/.

Singh, M. and Venkatraman, M. April 2001. Community-based service location. *Communication of the ACM*.

SUN. Visited May 2001. Java developer connection center. http://developer.java.sun.com/developer/products/jini/.

VD HEUVEL, W. 2001. Service representation, discovery, and composition for e-marketplace. *Cooperative Information Systems*. forthcoming.