Abstract
This paper presents the E-Commerce through Wireless Devices (E-CWE) project that uses software agents in the design and development of an m-commerce environment. Users of the E-CWE environment are able to use their wireless devices, such as mobile phones, to undertake e-commerce transactions. Conducting these transactions raises new issues that vary from the limited bandwidth of networks to the small size of screens. To support such users, software agents act on their behalf. For instance, the agents search for the products, negotiate with providers, and last but not least keep users informed about their progress. The operations that these agents carry out constitute itineraries. A technique to specify itineraries in the E-CWE environment is also discussed in this paper.

Author
Zakaria Maamar (zakaria.maamar@zu.ac.ae) is an associate professor at the College of Information Systems of Zayed University in Dubai, UAE. His research interests are related to software agents, wireless systems and mobile services. He received his MSc and PhD in computer sciences from Laval University in Quebec, Canada in 1995 and 1998 respectively.

Introduction
With the current development of information technologies, e.g., the World Wide Web, users are provided with more opportunities to conduct business over the Internet. For instance, there exist several websites that allow users to order their products online. This way of doing business constitutes a part of what is known as electronic commerce (e-commerce) (IBM 2001). In fact, Web shopping is only a small part of the e-commerce picture. The whole picture covers several types of businesses, from customer-based retail sites like Amazon.com (business-to-consumer), through auction and music sites like eBay or MP3.com, to business exchanges trading goods or services between corporations (business-to-business) (E-Commerce FAQ 2002). E-commerce has to be seen as a general term for any type of business, or commercial transaction, that involves the transfer of information across the Internet (E-Commerce FAQ 2002).

In (Froehlich et al. 1999), the authors asked ‘what makes e-commerce different from normal commerce?’ Their question was motivated by the claim that, since currently computers mediate almost all business transactions, all commerce is electronic. They answered that e-commerce is not just the presence of computers or absence of paper, but it implies more such as: using a non-proprietary open network; not requiring proprietary client software (i.e., any browser should do); servicing 24 hours a day, 7 days a week; and finally establishing the identities of parties without requiring physical contact.

Conducting e-commerce is still not largely accepted for different reasons. First, users have to search for the websites that offer online access to catalogues. Second, users have to understand how these sites operate. Third, users have to specify their needs according to the features of these sites, e.g., terminology. Fourth and for comparison needs, users may have to retain information while switching from one site to another. Last but not least, users have to be aware of security problems when submitting sensitive information, e.g., credit card numbers.

To assist users in conducting e-commerce transactions, we suggest associating them with Software Agents (SAs) (Maamar 2002) that will act on their behalf.

It is obvious that carrying out operations from a high bandwidth-connected workstation is more efficient and comfortable than carrying out the same operations from intermittently connected wireless devices. However, it may happen that a person would like to buy a gift for her daughter while travelling home by train. Using her mobile phone, the person accesses online stores looking for
the perfect gift (it is assumed that this person’s phone is WAP-enabled (cf. the second section). Her search is constrained by the maximum price she is willing to pay, the delivery time, and the age of her daughter. All the above-cited operations have to be initiated from the mobile phone of this person. Therefore, it becomes urgent to support persons in this kind of scenarios. This is the aim of the E-Commerce through Wireless Devices (E-CWE) project. Mobile commerce (m-commerce) is the new trend and is expected to drive the future development of e-commerce. Being able to buy and sell goods and services directly from wireless devices is a big step towards achieving ‘anywhere, anytime’. Location and time will no longer constrain persons from completing their operations. However, a new set of issues and impediments that are associated with m-commerce need to be dealt with differently (Tarasewich and Warkentin 2001).

In this paper, the following aspects are discussed according to the principles and features of the design of the E-CWE environment. How and why to assist users with SAs? How to deal with the characteristics of wireless devices, for example limited resources? How to offload computing from these devices to other platforms? How do the operations that SAs perform constitute itineraries to be specified? How to make these operations secure? (Tang and Veijalainen 2001) state that many mobile-device users are eager and ready to pay for advanced features, like wireless web access and location-oriented services. The various features of the participants (e.g., consumers, providers, brokers, delivery companies) that are involved in an m-commerce scenario, namely heterogeneity, physical distribution and ability to autonomously perform tasks but also to collaborate with other participants to achieve a common goal, highlight the suitability of SAs for modelling such a scenario. As it will be explained in the next section, an agent has several attributes that match the above-cited features.

The remainder of this paper is organized as follows. The next section presents an overview of software agents, wireless environments, and related work to m-commerce. This is followed by an introduction to the E-CWE environment architecture and its method of operating. The way providers contribute to the satisfaction of users’ needs is also, discussed in this section. We then discuss how security is addressed in the E-CWE environment. The final section provides some conclusions and discusses the implementation work currently being undertaken. Even if negotiation is among the important issues in e-/m-commerce projects, it does not currently fall in the scope of the paper.

PRELIMINARIES

The purpose of this section is to present briefly the main concepts that are used for the design of the E-CWE environment. Similar projects to E-CWE are, also, discussed.

Software agents (SA)

An SA is a computer program that is able to act autonomously and to adapt to its environment changes in order to reach its objectives (Jennings et al. 1998). The agent receives stimuli from its environment and acts in consequence. Being autonomous, the agent controls its actions and its internal state and acts without the intervention of other humans or agents. A SA exhibits a number of features that make it different from other traditional components: autonomous; goal-oriented; communicative; adaptive; mobile; collaborative; flexible; self-starting; temporal continuity; and character.

Wireless technologies

Several types of wireless devices exist, e.g., mobile phones and personal digital assistants. These devices suffer multiple limitations due to their batteries, autonomy and dimensions (Bergenti et al. 2001). These limitations suggest smaller CPUs, low memory, small displays and simplified input devices. The Wireless Application Protocol (WAP) is among the solutions that overcome these limitations. The WAP aims at supporting access to distributed services through wireless devices. Encoding documents for the needs of wireless devices requires representation languages of the data to be displayed. The Wireless Markup Language (WML) is an XML application developed in order to distribute content to such devices.

Related work to m-commerce

Several m-commerce applications are listed in Varshney and Vetter (2001), such as mobile financial applications (e.g., banking), mobile advertising (e.g., sending location sensitive advertisements to users), and mobile office (e.g., working in one’s car when caught in a traffic jam). Interesting are the requirements that are associated with each application. Samples of the requirements for the above-cited applications are (Varshney and Vetter 2001):

- Mobile financial applications: security, atomic transactions and support for disconnected operations.
- Mobile advertising: knowing the changing users’ needs, matching content to users’ preferences, and increasing content richness under limited bandwidth.
- Mobile office: security.

Besides these applications, various m-commerce initiatives are reported in WELCOM (2001). (Kannan et al. 2001) define wireless as anytime, anywhere and as needed. In (Garcia-Reinoso et al. 2001), m-mall is introduced. M-mall servers know the location of users in real-time and push information into the handhelds of users, regardless of the technology of these handhelds. Different types of
m-mall services are suggested such as advertisement, notification and guidance. Another work on m-commerce concerns the use of SAs to improve security and convenience (Tang and Veijalainen 2001). Two weaknesses are associated with m-commerce: (1) due to its small size, a mobile device can easily get lost and/or stolen, and (2) it is neither very convenient nor cheap for a mobile user to perform transactions that require scrolling several messages, scanning catalogues, or negotiating with partners. To handle these weaknesses, SAs are used for securing and increasing adaptability of m-commerce transactions. These transactions are made secure provided the wireless device has the capability of performing cryptographic algorithms (e.g., using the device-embedded smart card). Another work in m-commerce is called Agent Portal E-commerce (APE) (Mihailescu et al. 2000). APE is a Java-based application layer system that is designed specifically for users of wireless devices. APE uses mobile SAs to search and bargain on behalf of users for the best price for products or services offered by e-store fronts on the web.

THE E-CWE ENVIRONMENT

Architecture

In the E-CWE environment, we recommend the use of SAs for the following reasons:

- Interests and preferences of users are different. A static description of the required operations and involved resources that are generally spread across the net is not feasible. It is accepted that SAs cope with dynamic and distributed environments (Jennings et al. 1998).
- Autonomy allows SAs to act on users’ behalf for instance when time to make decisions comes.
- Mobility allows SAs to cope with the constraint of network bandwidth (Lange and Oshima 1999). An agent can move from one site to another. Once it arrives, the agent undertakes its operations locally.
- Last but not least, m-commerce is an environment rich with information making the use of SAs natural (Weilang and Huanye 2000). This richness is due to the several types of information (for example on products, providers and negotiation strategies) that need to be managed and filtered. SAs have definitely a great role in assisting users.

Figure 1 illustrates the architecture of the E-CWE environment. This architecture illustrates providers that have decided to offer online access to their services. Web portals are among the solutions to support this access. Providers are associated with different types of sites that range from basic ones (e.g., static catalogues) to complex ones (e.g., auction markets). In the E-CWE environment, users are associated with user-agents to whom their operations are entrusted. In the same figure, a repository of services exists. It describes services in terms of type, input parameters, output parameters, cost, and invocation method. Associated with provider-agents, providers subscribe to the repository of services. In Figure 1, the dashed lines represent wireless interactions. These interactions are of two types. The first type corresponds to users browsing the repository of services to identify the relevant providers. The second type corresponds to invoking the services of providers once these providers have been identified. We recall that all these interactions are transparent to users and providers, as well.

The role of user-agents is to satisfy the needs of users. User-agents convert needs into requests, browse the repository of services, identify the services from the repository that match with these requests, and finally design itineraries to specify the sequence of services to be invoked. Guidelines of how to design a sequence of services are given in the ‘Itinerary definition’ section. Provider-agents act on behalf of providers. Each provider-agent knows how to map the incoming messages of a user-agent into requests a provider understands. At the same time, the provider-agent knows how to map outgoing messages of the provider into a way user-agents understand.

Repository of services

The repository of services is the backbone of the E-CWE environment. It has an information-brokerage role; it supports the matching between needs of users and services of providers. A supervisor-agent is responsible of the repository of services; it consists of two types of services: user-services and provider-services. User-services are offered to users and give an overview of the needs that are satisfied. Samples of user-services are stock quotes monitoring and travel planning. In order to be satisfied, user-services require the composition of provider-services. Each provider subscribes to the repository in order to announce its provider-services (cf. Figure 1). In the rest of this paper and for the sake of simplicity, I-services will denote provider-services. Examples of provider-services
can be airline booking, accommodation booking, and car renting. This short list of provider-services can be constituted into a user-service.

The repository of services belongs to a working infrastructure (cf. Figure 2). It is a platform that manages user-agents and I-services at the following levels: creation, execution, communication, and migration if needed. The use of the working infrastructure also enables offloading a part of the computing operations from wireless devices to that infrastructure. The rationale is due to the limited resources of such devices. Furthermore, the working infrastructure permits conducting operations in a common and secure place avoiding thus, wireless connections. Security is achieved because all the interactions between user-agents and I-services are local; no remote interactions are needed so no risk of intercepting messages. Among the operations that are undertaken in the working infrastructure, we cite looking for relevant providers, interacting with providers, and planning for provider sites to visit. Details about all these operations are given in the next section.

Operating

Membership is required for users who aim to join the E-CWE environment. The same membership approach is suggested in (Mihailescu et al. 2000). Initially, a user of the E-CWE environment downloads from the repository of services to his mobile device the list of user-services that exist and saves them for later use. Transferring the list of user-services avoids access to the repository of services and hence, to connect to networks. Each time the list of user-services is updated (e.g., a new user-service is added), the list is broadcast to the E-CWE members. SMS messages are perfect for informing members about the availability of a new list of user-services to download. Operation (0) of Figure 2 illustrates the membership.

Figure 2 represents the operation of the E-CWE environment. In what follows, numbers between brackets correspond to the chronology of operations. First of all, the user browses the user-services that are available; he displays them on his mobile device (1). According to his needs, the user selects a user-service. Next, he specifies his needs and transfers them to the supervisor-agent of the repository of services (2). Finally, the user disconnects from the network and waits for a response to his request. Users’ needs are classified into must-have needs (i.e., requirement) and would-like-to-have needs (i.e., optional) (Marathe and Diwakar 2001). The second type of need is used to relax certain constraints if, for example, the search for relevant providers does not succeed. Through the interface, the user is responsible of identifying ‘must-have’ needs vs. ‘would-like-to-have’ needs. In travel planning, examples of must-have needs could be budget limit and number of people. Examples of would-like-to-have needs could be stay duration and transportation type. As soon as the supervisor-agent receives the user’s needs, it assigns a user-agent to satisfy them. While it is not represented in Figure 2, the supervisor-agent manages a bank of user-agents.

The next step transmits the user-agent from the bank of user-agents to the reception platform in which its installation is initiated (3). Once the user-agent is within the reception platform, it starts browsing the repository of services looking for the I-services that are required to satisfy the user’s needs (4) (the matching technique is not part of the E-CWE’s objectives). When the relevant I-services are identified, the user-agent requests the supervisor-agent to install them in the reception platform (5). Interactions between user-agents and I-services lead into signing contracts (6). For security reasons, I-services notify the provider-agents about these contracts (7). A contract specifies the names of the participants, the service that is involved, and the agreed cost. Additional elements can be added to a contract definition, if needed.

After signing contracts between user-agents and I-services, three exclusive situations take place in the E-CWE environment (cf. Figure 3). Letters a and b of Figure 3

---

**Figure 2. Operating the E-CWE Environment**
Figure 3. Situations in the E-CWE Operating Environment

denote the first two situations. The third situation combines situations a and b.

- Yes-visit situation (cf. Figure 3 -a-): the user-agent has to move to the provider sites in order to get all the information it needs. For instance, if the provider is an auction site, so it is important that the user to be in that site. Hereafter, we resume our description according to the chronology of Figure 2. When the itinerary of provider sites to visit is established (see next section), the user-agent migrates from one site to another (8.a), interacting with provider-agents (9.a). Finally, the user-agent sends results to user (10.a).

- No-visit situation (cf. Figure 3 -b-): the user-agent does not have to move to provider sites. Hereafter, we resume our description according to the chronology of Figure 2. After receiving the contracts from their I-services, provider-agents check these contracts, implement appropriate actions, and finally send results to user (8.b).

- Yes&No-visit situation (cf. Figure 3 -c-): a user-agent has to move to certain provider sites to meet with their provider-agents (cf. Situation a). In addition, other provider-agents from the remaining provider sites (those that will not be visited) will send to the user-agent the information it needs. This is based on the requests provider-agents have received from their respective I-services (cf. Situation b). In fact, provider-agents push information to the provider sites in which the user-agent is currently running.

Itinerary definition

To explain the importance of defining itineraries that user-agents will have to follow, we consider a trip-planning example: (i) a user wants to book a domestic flight and accommodation; (ii) he also wants to find some attractions to visit; and (iii) he would like to rent a car if the major attraction is far from the booked accommodation. This sequence shows that at least three providers are involved, i.e., airliner, accommodation and attraction. For instance, the services of the first provider can be requested according to yes-visit situation. The services of the second and third providers can be requested according to no-visit situation. The way these providers contribute to the satisfaction of this example is associated with an itinerary.

In the E-CWE environment, user-agents satisfy the needs of users in collaboration with provider-agents. This collaboration takes two different forms that are:

1. Local collaboration: this occurs when a user-agent roams the net of provider sites; it visits their provider-agents individually and asks them to process its requests (cf. Figure 3 -a-).
2. Remote collaboration: this occurs when a user-agent asks remote provider-agents to process its requests. However, instead of knowing how to interact with each provider-agent, the user-agent entrusts its requests to the I-services that act on provider-agents’ behalf (cf. Figure 3 -b-).

Local and remote collaboration forms have an impact on the itinerary that a user-agent has to follow to satisfy the user’s needs. This itinerary depends on the three above-mentioned situations.

We consider an itinerary as a set of interconnected services to be performed according to a specific form of collaboration. To specify the itinerary of a user-agent, we adopt a task-based representation of the services of that itinerary. In addition, to deal with the characteristics of the three above-mentioned situations, we enhance that representation with the following information: who participates in the execution of a service? How are the services
Table 1. Details on a Service Description

<table>
<thead>
<tr>
<th>Layer</th>
<th>Field/Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Previous service</td>
<td>Chronology at the itinerary level</td>
</tr>
<tr>
<td></td>
<td>Next service*</td>
<td>Chronology at the itinerary level</td>
</tr>
<tr>
<td></td>
<td>Performance type</td>
<td>Local (after mobility)/Remote</td>
</tr>
<tr>
<td>2</td>
<td>Diagram</td>
<td>Begin</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>End</td>
</tr>
<tr>
<td></td>
<td>E</td>
<td>Chronology at the service level</td>
</tr>
<tr>
<td></td>
<td>Task</td>
<td>Connection between tasks</td>
</tr>
<tr>
<td></td>
<td>Flow</td>
<td>If/loop structures</td>
</tr>
<tr>
<td>3</td>
<td>Data from previous service</td>
<td>0[in-data]n (what to expect)</td>
</tr>
<tr>
<td></td>
<td>Data for next service</td>
<td>0[out-data]n (what to forward)</td>
</tr>
<tr>
<td></td>
<td>Who will provide data?</td>
<td>0[agent, \to n]n (list of agents)</td>
</tr>
<tr>
<td></td>
<td>To whom data is provided?</td>
<td>0[\to agent, n]n (list of agents)</td>
</tr>
<tr>
<td></td>
<td>Participant user-agent</td>
<td>Is the user-agent involved (y/n)?</td>
</tr>
<tr>
<td></td>
<td>Provider-agent</td>
<td>Is the provider-agent involved (y/n)?</td>
</tr>
</tbody>
</table>

*Sonera (2002) suggests storing the secret key in the SIM card of the user’s mobile phone.

connected together? What data are exchanged between the services? Who provides data to a service? To whom are the data of a service provided? And, how is a service executed (remotely or locally)? Figure 4 illustrates the extended task-based representation of a service. This representation has three layers. Layer 2 corresponds to the execution chronology of a service. Meanwhile, layers 1 and 3 ensure the connection of that service to other services, i.e. pre and post-services. Layer 2 uses flowchart symbols.

Table 1 explains the elements that constitute the layers of Figure 4. The first column of Table 1 corresponds to layers; the second lists the fields/symbols used in each layer; and the third column describes each field.

When all the concepts needed to specify itineraries are set, we apply them to the situations that a user-agent meets when it implements itineraries. For clarification, we assume the existence of one user-agent, and three provider-agents, offering three services. To shorten our description and stay more focused, we limit our description to the yes-visit situation (cf. Figure 3-a-).

Yes-visit situation

User-agent needs service of provider-agent, and service of provider-agent. The itinerary of user-agent consists of first migrating from the E-CWE working infrastructure to the provider-agent site and then, to the provider-agent site coming from the provider-agent site. Figure 5 represents how this itinerary is specified. The different fields of Table 2 are instantiated according to the characteristics of the current situation. Table 2 shows the sequence of events.

E-CWE SECURITY

E-/M-commerce applications require the exchange of sensitive information that has to be secured. Several security issues need to be addressed: confidentiality, authentication, integrity and non-repudiation. In the E-CWE environment, we focus on authentication and confidentiality. In authentication, providers have to check the identity of
user-agents before they allow them to enter their sites. The identity check is preceded by contracts validation (cf. Figure 2, operation 6). In confidentiality, the payments of user-agents to the services they receive from providers have to be done in a secure way. We consider different technologies for meeting both requirements. SPKI (Simple Public Key Infrastructure) is the one that interests us (Ellison et al. 2001); it relies on publishing public-keys to potential participants.

In the E-CWE environment, security goes through two steps: secure service access and secure payment. Our emphasis in this paper is on the first step. Providers (here, we refer to provider-agents) authenticate users (here, we refer to user-agents) by exchanging certificates. A provider signs a certificate using its private key (1). Afterwards, he sends the certificate to a user who verifies it using the provider’s public key. Then, the user signs the certificate using his secret key (2) (if using several services filling in the ‘Next Service’ field will mean that the execution of those services will be done in parallel). Finally, the user returns the certificate to the provider who will verify it using the user’s public key (3). Figure 6 outlines the authentication step between users and providers. We denote a public key by Kp and a private key by Ks; their owners tag these keys. In this figure, M stands for message that contains any data. Let us point out that operation 1 of Figure 6 is already supported by WAP. Authentication should also take place between users and payment systems as well as between providers and payment systems (Figure 6, operations 1 to 3).

CONCLUSION

In this paper, we presented the major characteristics of the E-CWE environment. It aims at designing and developing a software agent-based m-commerce environment. To this end, we suggested two types of agents and a working infrastructure. Among these agents, user-agents were able to get users’ needs, convert them into requests, search for relevant providers, interact with these providers, and last but not least move to provider sites according to the needs to be satisfied. The working infrastructure had an important role to play in offloading the computing of certain operations.
from users’ devices to that infrastructure. At the security level, the E-CWE environment presents authentication, confidentiality, and integrity aspects. Certificates such as SPKI achieve authentication. Private/public keys achieve confidentiality. And, hash-coding functions achieve integrity. Integrity makes sure that exchanged information is not altered. Another security aspect in E-CWE is the non-repudiation property. Neither the user nor the provider can deny the messages they sent. Encrypting messages with private keys supports this property.

Several elements of the E-CWE project provide scope for further analysis. Among them, we intend to work on user’s spatial location. In fact, it would be interesting to know the user’s direction of travel at any given instant of time. For example, our fictional person with her daughter’s gift may desire to contact the toy store that is on her way home or the store nearest to her current position. To this end, the user can specify for example her direction as well as how long approximately it takes to reach her destination. Another activity consists of introducing device-agents to reside in users’ wireless devices. Device-agents will be in charge of displaying results that are obtained from provider-agents according to the display features of each wireless device, transferring users’ needs to the supervisor-agent, monitoring the state of wireless device’s batteries, and beeping the supervisor-agent in case there is a delay in returning the user’s response.

ACKNOWLEDGMENTS

The author would like to thank the reviewers for their useful comments as well as Hamdi Yahyaoui from Laval University, Canada, and Wathiq Mansoor from Zayed University, UAE, for their contribution to the E-CWE project.

References


