Automating negotiation for m-services.

Shamimabi Paurobally[1] *
Phillip J. Turner[1]
Nicholas R. Jennings[1]

[1] School of Electronics and Computer Science, University of Southampton

* Shamimabi Paurobally is now based within the School of Informatics, University of Westminster

Automating Negotiation for M-Services

Shamimabi Paurobally, Phillip J. Turner, and Nicholas R. Jennings

***Abstract*—Mobile electronic commerce (m-commerce) is an emerging manifestation of internet electronic commerce that bridges the domains of Internet, mobile computing and wireless telecommunications in order to provide an array of sophisticated services (m-services) to mobile users. To date, much of the research in the area has concentrated on the problem of service discovery. However, once a service has been discovered, it needs to be provisioned according to the goals and constraints of the service provider and the service consumer. Since, in general, these will be different stakeholders (with different aims), the *de facto* provisioning method will be some form of negotiation. To this end, this paper develops automated negotiation protocols and strategies that are applicable in m-commerce environments. Specifically, we develop and evaluate time-constrained bilateral negotiation algorithms, that allow software agents to adapt to the quality of the network and/or their experience of similar interactions.**

***Index Terms*—Agents, automated negotiation, m-services.**

I. Introduction

FUTURE generation mobile telecommunication systems are increasingly being viewed as open market places in which the various stakeholders produce and consume services [1]. This view yields a convergence of electronic commerce, wireless networks and the Internet. Examples of such m-services include mobile shopping (e.g., Mr. Smith's software agent books a flight from a PDA, then reserves a rental car and a restaurant on his arrival), location-sensitive information (e.g., obtaining map services, local hotels, and weather information), telemetry (e.g., receiving traffic updates and logistics tracking) and mobile banking (e.g., billing of services, buying stocks and contacting banks through mobile devices).

As these examples imply, to be truly effective m-services have to be both customised and personalised, while being location-sensitive and context-aware. For example, in terms of location sensitiveness and personalization, en route to the airport Mr. Smith's agent might receive updates on the traffic routes, road works and weather information. Context-awareness would, for example, customise the information display according to whether Mr. Smith has a passenger who is helping him analyze the traffic and weather updates (e.g., if Mr. Smith is alone, then only keywords would be displayed in a larger font, otherwise if there is a passenger then more details would be given, along with a map and landmarks, for the passenger to read and analyze). In terms of customization and personalization, Mr. Smith may either prefer to be presented with traffic updates for the whole route to the destination or only local information updates for the next two miles. Moreover, these updates may take into account Mr. Smith's driving skills (e.g., driving in the fast or slower lanes on the motorway) and his preferences of how to present the information (e.g., graphically and/or as audio).

In order to offer m-services with such properties and, at the same time, to be effective at the speeds and capacities demanded in wireless systems, the processes of service discovery, service provisioning and service execution need to be automated to a significant extent. Since these operations have to be performed in a highly dynamic, uncertain and unpredictable environment it is important that the software systems can act and interact in flexible ways in order to achieve their objectives. To this end, the individual service producers and consumers can naturally be viewed as software agents (since these are exactly the properties and types of environment that lend themselves to an agent-based approach [2]). Specifically, these agents are able to meet their design objectives by having control over their own behavior (autonomy), having the ability to respond to changes in their environment (reactivity), and the ability to act in anticipation of their aims (proactivity) [3]. Now, since the agents are autonomous and because they represent different organizations, with different aims and objectives, the *de facto* means by which they will interact is some form of negotiation [4] (here defined as a form of decision making where two or more agents jointly search a space of possible solutions with the goal of reaching a consensus).

Against this background, this paper investigates the requirements and mechanisms for automated negotiation for m-services in a m-commerce environment. In particular, we focus on bilateral negotiations (since these are common in such environments [5]) and given our aim to deploy such systems we focus on the performance aspect of the negotiations. In more detail, we first discuss the characteristics of wireless communications that effect agent negotiations. Taking into account the quality of the underlying communication network and an agent's interaction experiences, a bilateral protocol is developed and combined with decision-making mechanisms for evaluating and generating exchanged sets of negotiation issues. We then evaluate these algorithms with respect to the key performance metrics for this domain.

This paper advances the state of the art by proposing an automated negotiation facility that allows agents to adapt to their prevailing situation in mobile telecommunication environments. From the network's perspective, our work extends current research in software-based mobile telecommunications to more sophisticated forms of multiissue negotiation for the provision of personalised services. From an agent-oriented perspective, we design automated negotiation mechanisms to cope with the limitations imposed by the underlying communication

Manuscript received January 31, 2002; revised March 23, 2003. This work was supported by Mobile Virtual Center of Excellence in Mobile and Personal Communications including that of EPSRC. This paper was recommended by Guest Editors Z. Maamar and B. Benatallah.

The authors are with the School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K. (e-mail: sp@ecs.soton.ac.uk; pjt@ecs.soton.ac.uk; nrj@ecs.soton.ac.uk).

Digital Object Identifier 10.1109/TSMCA.2003.819654

infrastructure. In particular, the objective is to demonstrate that automated negotiation permits a high degree of flexibility in establishing new m-services. Toward this end, we extend the alternating offers protocol that is normally used in time-constrained settings [6]–[8] to incorporate an interaction state. For example, in the stateless alternating offers protocol, it is always possible to perform offers and counter-offers, to make agreements and send rejections. In constrast in our protocol, an agent has to take into account previous actions and past states in order to infer the next possible actions and states. Morevoer, the time, experience and network aware negotiation strategies we develop can also refer to previous, current and allowable future sequences of interaction states in order to determine how to respond. Thus, for different negotiation threads, similar sequences of states may yield different results depending on the run-time dynamics. Such adaptation to limited resources is important in m-commerce environments because there are inevitably strong resource-bounds and a high degree of variability in the underlying infrastructure. When taken together, these characteristics mean adaptation is essential if an agent is to effectively achieve its goals.

The remainder of the paper is structured in the following way. Section II relates the m-service and the agent paradigms. Section III discusses the various technologies and requirements for automating m-service negotiation. Sections IV and V specify the bilateral negotiation mechanisms – covering both the protocol and the decision making algorithms. Section VI discusses the performance of the algorithms with respect to the identified set of metrics. Section VII summarizes related work and Section VIII concludes.

II. M-Commerce and Software Agents

M-commerce is concerned with the set of applications and services accessible from Internet-enabled mobile devices [1]. It has a number of requirements over and above those of more traditional e-commerce including services that are accessible over wireless networks and that are adaptable according to the characteristics of the mobile devices for which they are configured and on which they are run [9].

In this area, I-Mode [10] by NTT DoCoMo is an example of an early m-commerce application, offering wireless web browsing and e-mail from mobile phones, where the users are charged according to the volume of data transmitted. As the technology progresses, location-sensitive and context-aware services will become a more routine part of the offering. For example, users of location-sensitive devices will be able to search for directions to nearby restaurants, banks and similar listed events in their area. Moreover, the limited screen size, low data rates associated with mobile Internet devices, rapid deployment of accurate location-tracking technology, as well as the time critical nature of many of the tasks in which mobile users engage, are all likely to contribute to the increasing demand for mobile location-sensitive services [1].

As it currently stands, the limitations of the mobile devices constrain the accessibility of m-services. Therefore, m-commerce does not support the full hyper-, multimedia found on the wired Internet and requires well-targeted and concise content presented to the user. For example, animations, banners and lists of results from search engines are difficult to present on mobile phones. For this reason, customers need to be presented with services that are relevant to their current locations, preferences and activities. This, in turn, requires smarter interfaces and applications that learn from users' behaviors [11].

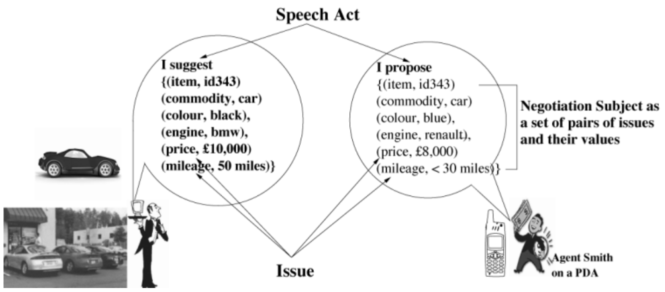

Fig. 1. Negotiation by exchanging speech-acts and a negotiation subject.

To achieve such personalization, negotiation can be used in the trading of both telecommunications services and high-level services between participants in an open electronic market. As an illustration of the former, consider the case of Mr. Smith who routinely undertakes a train journey. The agents on his PDA may negotiate for bandwidth so that he can watch the news, without interference from other used frequencies, as he travels. Thus, the agents can provide smart and dynamically configurable networks for increased performance and robustness, foreseeing faults and changes in the environment. As an illustration of the latter, Mr. Smith's agent may learn the types of films and documentaries that he likes to watch and (bilaterally) negotiate to receive such multimedia presentations from different content providers. The agents may also learn how Mr. Smith would like the MPEG-4 files to be displayed (e.g., full-screen, brightness and sound-level).

Achieving this vision is a difficult task. Wireless networks present a significant challenge for automating agent interactions because such mobile communications fall prey to low bandwidth, bounded coverage, latency, error rates and spurious connections (as discussed more fully in Section III-B). For this reason, negotiation models that are specifically tailored to the m-service domain need to be developed.

III. Automated M-Service Negotiation

As discussed previously, negotiation is fundamental to the provision and management of m-services in a marketplace model. However, automating this negotiation is a challenge that requires the following components to be exploited.

- *Agent languages* [12], usually in the form of speech-acts [13], specify the structure of exchanged messages and performatives (see Fig. 1).
- *Protocols* define the norms that govern a negotiation. They specify the actions (sequences of messages) permissible by an agent leading to some state [14]. If all agents comply with the same interaction protocol, they can expect certain responses from others and carry out a conversation.
- A group of agents negotiate about a *set of issues* called a *negotiation subject*. This set of issues can be in the

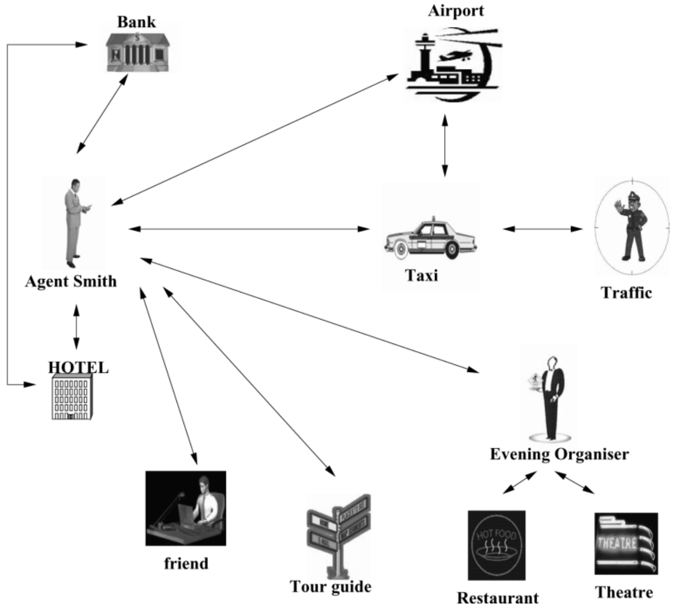

Fig. 2. M-commerce scenario involving negotiation for a travelling customer.

form of name-value pairs, e.g., renting a car with attributes *{(commodity, car), (color, black), (engine, bmw), (price, 10 000)}* as in Fig. 1. At the beginning of a negotiation, the set of issues each agent sends to the other may not fall in the intersection zone of their preferences (meaning an agreement is not possible). Usually, the participants can modify the issues and their values, for example, when evaluating and generating responses in an interaction. Negotiations terminate when the set of issues satisfies all the agents' preferences (an agreement is reached) or when one of the parties teminates the encounter (for whatever reason). For the sake of conciseness, in the rest of this paper, we refer to values for sets of issues as sets of issues or the issue set.

- Each agent needs an *evaluation function* for a negotiation subject that reflects the agent's preferences. An agent chooses its *strategies* privately for evaluating, generating and deciding on its next course of action. An agent uses some decision process to determine its positions, concessions and criteria for agreement and since it is self-interested, it will choose the strategy that gives it the best return.

A. Automated M-Service Negotiation Scenario

Having defined the basic building blocks of automated negotiation, this subsection examines in more detail the negotiations involved in a typical m-service scenario (Fig. 2). In this context, the labels j_i on the arrows represent a particular order of exchanges between the participants. Thus, information flows between the entities as they negotiate with each other for services. Here Mr. Smith wishes his agent on his PDA to plan his New York trip while he travels to and around the city. As Mr Smith moves through different wireless networks, his agent has to adapt its negotiation behavior to the quality of the service of the prevailing network. Specifically, the interactions involving Agent Smith are shown in Table I.

TABLE I
INTERACTIONS IN THE TRAVELING CUSTOMER SCENARIO

j_i	**Interaction Flow**
j_1	On arriving in New York, Agent Smith requests the airport agent to book a taxi to go to the hotel.
j_2	The airport agent negotiates with agents from taxi companies for an available taxi to drop Mr. Smith to his hotel. The taxi company debits the fare from Mr. Smith's bank account.
j_3	The taxi picks up Mr. Smith after the taxi agent identifies Mr. Smith from the details uploaded by Agent Smith.
j_4	The taxi agent dynamically downloads the map and best route from a traffic monitoring agent. At the same time, the taxi agent offers various radio channels, depending on their location, to Agent Smith for Mr. Smith to listen to the news via his headphones.
$j_{5(i)}$	Whilst in the taxi, Agent Smith informs the hotel agent of its imminent arrival and negotiates room rates.
$j_{(5(ii)}$	While Agent Smith is negotiating the room rates, it is concurrently negotiating with its bank to transfer cash from Mr. Smith's bank account into local currency (which is to be picked up at the nearest American Express branch). On arriving at the hotel, Agent Smith checks in with the hotel agent, uploads its credentials and downloads fire instructions, breakfast and checking out times.
j_6	The hotel checks that Mr Smith has enough funds to cover his hotel bill.
$j_{7(i)}$	There is an open network of online agent platforms hosting diverse agent-based services in New York, available via an evening organiser. Agent Smith contacts and negotiates with the evening organiser for eating and entertainment services.
$j_{7(ii)}$	En route to the theatre, Agent Smith concurrently negotiates with the evening organiser for the play Mr Smith is going to watch and the seating arrangements. Agent Smith also investigates a sight-seeing tour for the next day.
$j_{7(iii)}$	While negotiating for the sight-seeing tour, Agent Smith negotiates with Mr Smith's friend to accompany him. They deliberate on when and what they wish to visit in New York. Agent Smith has to take into account the wireless communications between the evening organiser, the sight-seeing company and Mr. Smith's friend.

In more detail, Fig. 3 shows an instance of a conversation between Agent Smith and other parties, while Mr. Smith is travelling in the taxi to the restaurant. In particular, we show a subset of the messages sent in the three concurrent negotiation threads, $j_{7(i)}$ to $j_{7(iii)}$ in Fig. 3. In this situation, Agent Smith is simultaneously negotiating with the evening organizer about theatre tickets, a tour guide for a visit of New York on the next day, and with a friend, (friend$_1$) to accompany him on the visit. Whilst Mr. Smith is in the taxi, the quality of his wireless connection via his PDA changes with migration between networks (covering the business district, the old town and the suburbs). To this end, let us consider the more detailed message exchange when the taxi is travelling in the old town (centre box in Fig. 3). At this point, the performance of the network is around average and this allows Agent Smith to conduct simultaneous negotiation threads with the evening organizer, the tour guide and his friend$_1$.

- Message exchange between Agent Smith and the evening organizer.
  - $j_{7(i)a}$: The evening organizer proposes to Agent Smith tickets to watch the play Othello at $70.
  - $j_{7(i)b}$: Agent Smith can afford to bargain and replies with another proposal to watch the play Hamlet, with balcony seating at a price of $30.

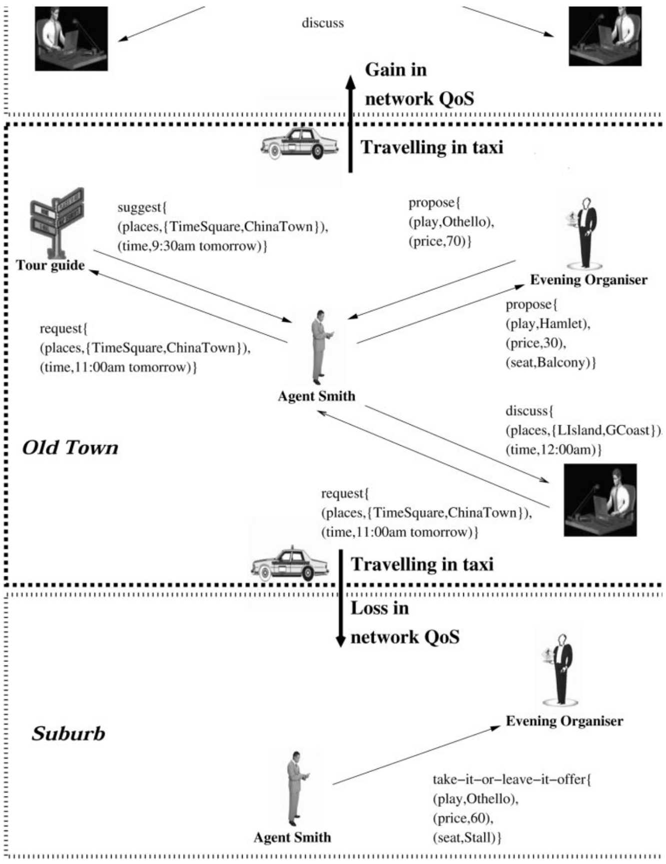
discuss
Gain in network QoS
Travelling in taxi
suggest{ (places,{TimeSquare,ChinaTown}), (time,9:30am tomorrow)}
Tour guide
propose{ (play,Othello), (price,70)}
Evening Organiser
propose{ (play,Hamlet), (price,30), (seat,Balcony)}
request{ (places,{TimeSquare,ChinaTown}), (time,11:00am tomorrow)}
Agent Smith
discuss{ (places,{LIsland,GCoast}) (time,12:00am)}
Old Town
request{ (places,{TimeSquare,ChinaTown}), (time,11:00am tomorrow)}
Travelling in taxi
Loss in network QoS
Evening Organiser
Suburb
Agent Smith
take–it–or–leave–it–offer{ (play,Othello), (price,60), (seat,Stall)}

Fig. 3. Instance of a message exchange in the travelling customer scenario.

- Message exchange between Agent Smith and the tour guide.
  - $j_{7(ii)a}$: The tour guide suggests to Agent Smith a visit to the Times Square and China Town at 9:30 the next morning.
  - $j_{7(ii)b}$: Agent Smith also chooses to bargain with the tour guide and requests a visit to the Times Square and China Town, but at 11:00 the next morning.
- Message exchange between Agent Smith and friend$_1$.
  - $j_{7(iii)a}$: Friend$_1$ wishes to discuss a visit to Long Island and the Gold Coast at noon the next day.
  - $j_{7(iii)b}$: A gent Smith instead proposes to friend_1 to accompany him to the Times Square and Chinatown at 11:00 the next morning.

As Mr. Smith's taxi travels in the suburb (lower box in Fig. 3), the quality of the network decreases and there is a loss of performance in the communication layer underpinning Agent Smith's conversations. As a consequence of this degradation, Agent Smith suspends its negotiation threads with the tour guide and his friend for visiting New York the next day (because they are less urgent). Agent Smith continues negotiating with the evening organizer, but to expedite the process it sends an ultimatum (take it or leave it offer) to buy Othella theatre tickets for $60 with seating in the stalls. Thus it can be seen that the environment limits Agent Smith's resources and the agent has to respond accordingly (by decreasing the number of concurrent negotiations, conceding more quickly or reaching an agreement fast).

On the other hand, if Mr. Smith's taxi travelled to the business district (top box in Fig. 3), it may experience a more efficient and reliable network connection. In this case, Agent Smith can not only keep its existing negotiation threads, but may also increase its number of negotiations and decide to bargain more. Hence, Agent Smith might discuss and bargain more with the evening organizer to obtain an even better deal. For example, Agent Smith can now download bigger files faster, and may ask for an MPEG presentation of New York from the tour guide and may wish to discuss in more details the tour for the next day. In addition, Agent Smith may now negotiate with *two* friends concurrently, instead of sequentially, so as to plan the rest of his stay in New York.

In summary, the scenario shows that there can be many different parties that need to negotiate. These negotiations are often bilateral and they may also be concurrent. Each negotiation has its own parties, subject of negotiation, history and current negotiation state. In addition, each negotiation is differently influenced by environmental factors such as the quality of the network, the market dynamics or the neighboring agents.

B. Wireless Operational Factors Effecting Negotiation

In earlier generation networks, low data rates and long connection set-up times in mobile devices gave rise to concerns for optimization of the quality of the network (especially when persistent connections were needed for downloading from the Internet). Currently, some of these problems have been alleviated, but nevertheless mobile and fixed hosts still have different constraints in terms of power supply, latency or available memory. Therefore, since the agents have to operate in such environments, these factors need to be taken into account when designing the system (in general) and its interactions (in particular). In our context, this means different negotiation algorithms have to be specified for different situations. For example, it makes sense for mobile devices to be involved in short negotiations whereas fixed hosts can take part in continous and computationally expensive encounters.

Given this, when designing negotiation mechanisms for mobile environments, the peculiar characteristics of wireless devices and networks must be considered. In particular, the constraints of mobile telecommunications are often inter-related where the quality of service (QoS) may itself be parameterised with the other characteristics of the network (e.g., bandwidth, range, frequency of disconnections, costs of connection, data integrity, and security) [15]. For example, the variation in bandwidth and QoS of the underlying network means negotiation mechanisms should adapt to the varying environment conditions so as to adopt compensating actions and still find mutually acceptable agreements. That is, the QoS of the network could be used to influence the rates of concession and the decisions of an agent of whether to agree to a suboptimal deal. For example, if the QoS is low then an agent might agree to the first acceptable offer, while if the QoS is high, then it may try to bargain, search for the best deal and maximize its profit. Bandwidth limitations and fluctuations could also restrict the number of simultaneous users involved in a negotiation and the number of messages required to terminate the negotiation. For example, if the battery is running out, then an agent may decide to concede and quickly find an agreement or notify the others that it will soon suspend the negotiation. If the memory and processing power are particularly limited in a mobile device, then an agent can choose not to adopt complex strategies. In the case of increased latency and loss of network performance, an agent may choose to timeout or increase its time to compute its strategies and plans while waiting for a message.

C. Application Level Features Effecting Negotiation

There are also a number of features of the application (agent) level that effect the negotiation in this domain. While some of these features can also apply to wired networks, they tend to be more prominent in mobile telecommunications.

- *Scalability and availability*. The system must be able to cope with a large number of negotiations at any one instant.
- *Graceful Degradation*. Mobile communications systems are characterized by temporary failure, arbitrary or intentional transience of connectivity and reduced network performance due to resource shortages. In such situations, the negotiations must degrade gracefully with only bounded loss.
- *Fraud prevention and detection*. Robustness against fraud is essential if negotiation services are to be widely trusted and accepted. Thus, a malicious party (supplier or customer) should not be able to make significant fraudulent gains from repeated induced failures or by exploiting the fact that wireless communications are more prone to disconnections than wired ones. For example, an agent should not be able to negotiate to learn about the preferences of its opponents and then pretend that the wireless communication has failed just before an agreement is made. Knowing that its opponent is desperate for a service (e.g., its deadline is close), the agent then restarts another negotiation, after purposefully waiting for some time, and exploits the constraints of its opponent.
- *Deadlines*. Deadlines, as part of the resources of an agent, are important determinants of behavior. For example, an agent needs to consider whether its deadline is close and reach an agreement fast or whether it has enough time to bargain.

In summary, both the characteristics of the telecommunication and the market environment influence the choice of negotiation mechanisms and the behaviors of the negotiating agents. Given this, the next section develops negotiation mechanisms that are suitable for m-commerce.

IV. Negotiation Mechanisms

This section specifies mechanisms for negotiating for m-services between two agents. First, we specify a bilateral negotiation protocol as a pattern of high-level message exchange that two agents may follow in interacting with one another. The protocol may be regarded as the set of public rules or guidelines indicating the conduct of an agent toward other agents when carrying out the negotiation. Then, we design three possible strategies (adaptive to time, adaptive to time and quality of the network, and adaptive to past experience), within the same bilateral protocol. The agent designer is free to choose which of these strategies should be used for evaluating, generating and deciding on its next action or an agent may be designed to autonomously vary its strategies depending on its constraints and the environment. This choice is private to the agent.

1) Time-dependent strategy, $\mathcal{T}$, considers the deadline of an agent. This strategy may be extended to consider the resources of an agent, its opponent's behaviors or trade-off deals instead of successive concessions.
2) Network-aware strategy, $\mathcal{Q}$, allows an agent to take into account the variations in the quality of the network in its concessions.
3) Experience strategy, $\mathcal{E}$, takes into account the experience of an agent gained from previous and parallel negotiations, possibly with similar parameters (for example, the same opponents, similar subjects of negotiation, preferences and constraints).

We choose the above strategies for several reasons. Strategies $\mathcal{T}$ and $\mathcal{Q}$ are useful in m-commerce since an agent has time constraints and has to adapt to a varying network (as discussed in Section III). The strategy $\mathcal{E}$ emphasises an agent's social environment since information about past and current negotiations allows an agent to make informed decisions, minimize the risk of disagreements and avoid wasting resources (which are limited in m-commerce settings). The strategies may be combined and given weights to take into account the relative importance of various environmental aspects. For example, the two strategies $\mathcal{Q}$ and $\mathcal{E}$ may be combined to obtain a hybrid strategy $\mathcal{Q}+\mathcal{E}$ that considers both the quality of the network and the previous experiences of an agent. Such a hybrid strategy allows more flexible agent behavior, since more operational factors are considered. However, hybrid strategies also require more resources in terms of computational time, power and space and these may not always be available in m-commerce domains.

The rationale for the formulation of these strategies lies with our focus on automating negotiation for m-services rather than proposing new negotiation theories and analysis in m-commerce. Quoting from [16], "a key issue here (in multiagent systems) is that, since we are interested in actually building agents that will be capable of negotiating on our behalf, it is not enough simply to have agents that get the best outcome in theory—they must be able to obtain the best outcome in practice." Therefore we purposefully choose to develop practical strategies and to concentrate on how they adapt and learn from experiences in their environment. Thus, this paper is not concerned with a solely theoretical formulation and analysis of the strategies. Rather, we are concerned with how the negotiating agents evaluate and generate values for sets of issues, given sequences of states and a variety of available speech-acts.

A. Bilateral Protocol

Generally speaking, protocols are used to coordinate the activities of a group of agents as they try to satisfy their goals. In particular, this section specifies a protocol [17] between two agents looking for an agreement over a negotiation subject. The protocol allows requests, proposals, offers and agreements and may form the basis for further customization to allow richer interactions.

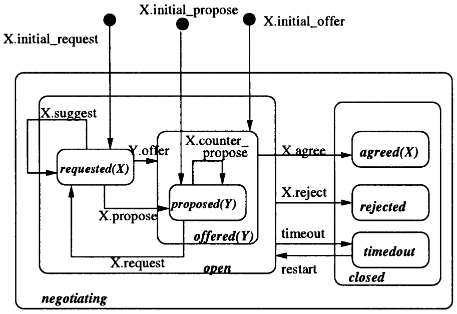

Fig. 4. Bilateral protocol.

In more detail, Fig. 4 gives the statechart of a bilateral protocol between agents *X* and *Y*. The protocol is portrayed in a statechart (instead of in other notations such as sequence diagrams or Petri nets because of the lack of expressiveness of the latter notations for showing multiagent interactions [14]).[1]

Entry in the negotiation is through either agent performing an *initial_request*, *initial_offer* or *initial_propose* message leading to an *open* state and more precisely to a *requested*, *offered* or *proposed* state respectively. For example, the process *X.initial_request* means that agent *X* sent an *initial_request* leading to a *requested(X)* state. The state *requested(X)* is read as agent *X* has triggered the state *requested*. Each state may be interpreted as conveying a level of commitment toward an agreement. For example an *agreed* state entails more commitment than a *requested* state. A *reject* action or *timeout* event can occur at any sub-state of an *open* state. From a *requested* state, both agents can continuously make *suggest* actions to remain in that state while modifying the subject of negotiation until one of them wants to move to a higher level of commitment through an *offer* or *propose*. The *proposed* state is a sub-state of *offered* and both of these states may allow an agreement to follow in the next action (leading to an *agreed* state). The difference between *offered* and *proposed* is that from the former state, an agent can only *agree* or *reject* whereas from the *proposed* state, an agent may *agree*, *reject* or return to a *requested* state through a request. We also allow the two agents to restart a *timedout* negotiation through forking into another bilateral negotiation with the negotiation subject being whether to restart the interaction.

B. Time-Dependent Strategy ($\mathcal{T}$)

This section specifies the decision making mechanisms for an agent according to its resources or its opponent's behavior (and is broadly based on [18]). The strategy depends on an agent's deadline or its reservation values. Decision theory [19] is used in this context as it is suitable for analyzing different alternatives under uncertain conditions and unknown outcomes of an action. As discussed earlier, the structure of a negotiation subject is a set of issue-value pairs. A rational agent aims to maximize its gain which depends on the result of an evaluation of the agreed set of issues. Such evaluation functions allow an agent to evaluate messages from other agents and to generate a new set of issues to respond with. As in [20], we consider a negotiation between two agents *a* and *b* over a changeable set of issues J. An issue j, ($j \in J$), can take values between $[\min_j, \max_j]$, which define the domain, D_j, of a quantitative issue and is effectively the reservation values of an agent. The domain of a qualitative issue is defined as an ordered set of possible values as in $D_j = \{q_1, \ldots, q_n\}$. Currently in our experiments, an agent's reservation values and the domain of the qualitative issues are provided by the designer. However, in long-running simulations, these values can be learnt by an agent via its experience strategy.

1) Evaluation Mechanisms: Similar to [20], let *a* and *b* designate two negotiating agents; let x be a set of pairs of issues and their values, as in { *(price,30), (quantity,2)*}. Let j designate an issue and let $x[j]$ be the value of the jth issue in the set x. Let the term $(x^t_{b\to a})$ denote the negotiation subject, a set consisting of values associated to independent issues, sent from agent *b* to *a* at time t.

Evaluation of $(x^t_{b\to a})$ involves summing the valuation (score) of each issue in the negotiation subject. The evaluation function of agent *a* about an issue j is given by V^a_j where $V^a_j : D_j \to [0, 1]$. A weight ω^a_j is associated by agent *a* to issue j where the sum of weights of all issues is 1. An agent can change its preferences for an issue by changing the weights associated to that issue. A set x, consisting of issues j, is rated by agent *a* as

$$V^a(x) = \sum_{1\leq j\leq n} \omega^a_j V^a_j(x[j]). \tag{1}$$

2) Offer Generation Mechanisms: An agent may adopt a behavior, according to its strategy and constraints such as time here, when generating a set of issues and in calculating how much to concede (e.g., it concedes more nearer to its deadline, as the level of its resource diminishes or as its opponent concedes). Here, we develop time-dependent, ρ_{time}, and opponent-dependent, $\rho_{opponent}$, behaviors. The following defines a time-dependent behavior. Let $t^a_{\max}$ denote agent *a*'s deadline. $(x^t_{a\to b})_{\mathcal{T}}$ denotes the set of issues, x, an agent *a* sends to agent *b* at time t, according to strategy $\mathcal{T}$ [18]. $(x^t_{a\to b})[j]_{\mathcal{T}}$ denotes the issue j in the set $(x^t_{a\to b})_{\mathcal{T}}$.

$$\begin{aligned}(x^t_{a\to b})[j]_{\mathcal{T}} =& \min{}^a_j + \frac{\min(t, t_{\max})}{t_{\max}}(\max{}^a_j - \min{}^a_j),\\ &\text{if } V^a_j \text{ is decreasing} \qquad (2)\\ (x^t_{a\to b})[j]_{\mathcal{T}} =& \min{}^a_j + \left(1 - \frac{\min(t, t_{\max})}{t_{\max}}\right)(\max{}^a_j - \min{}^a_j),\\ &\text{if } V^a_j \text{ is increasing.} \qquad (3)\end{aligned}$$

At the start of a negotiation, an agent does not normally concede and at its deadline the agent concedes to its reservation limit.

In the opponent-dependent behavior, an agent determines its concessions based on the previous attitudes of its opponents (from $\delta \geq 1$ steps ago).

$$x^{t_{n+1}}_{a\to b}[j]_{\mathcal{T}} = \min\left(\max\left(\frac{x^{t_{n-2\delta}}_{b\to a}[j]}{x^{t_{n-2\delta+2}}_{b\to a}[j]} \times x^{t_{n-1}}_{a\to b}[j], \min{}^a_j\right), \max{}^a_j\right). \tag{4}$$

[1] For example, in these notations, the roles are not bound to an agent's identity, timeouts and reject messages are hard to represent and, in their standard forms, there is no concept of an agent performing an action.

TABLE II
WEIGHTS ASSOCIATED TO TIME AND OPPONENT STRATEGIES

Time left	ρ_{time}	$\rho_{opponent}$
much_time_left	0.2	0.8
middle_time_left	0.4	0.6
less_time_left	0.8	0.2
no_time_left	1	0

The set of issues agent *a* generates at time t_{n+1} is within *a*'s acceptable values and proportionally imitates *b*'s behaviors. An agent generates its response by using a weighed combination of the time-dependent (ρ_{time}) and opponent-dependent ($\rho_{opponent}$) behaviors. The weights associated to the two behaviors depend on the amount of time left until an agent's deadline (ordered increasingly by the predicates *much_time_left, middle_time_left, less_time_left* and *no_time_left*). For example, if there is *no_time_left*, then an agent prefers the time-dependent behavior (ρ_{time}) so as to show more adaptation to the limited time. Whereas if there is *much_time_left*, an agent places more importance on the opponent-dependent behavior so as to gain more when time is not scarce. For example, Table II shows how the preferences of an agent for a strategy can be varied with the closeness of its deadline, where the weights are normalized and grounded in this case. Broadly speaking, this strategy shows that as time elapses, an agent's behavior becomes predominantly time-dependent.

For more general functions, at time t, the weights associated to time-dependent or opponent-dependent strategies follow simple decay functions with a rate of growth denoted by β. These functions reflect the relative importance placed on each strategy according to the time left, which is itself the difference between an agent's deadline and the current time. We use decay functions since they are simple to compute.

$$\rho_{time} = \frac{1}{(\beta t + 1)} \tag{5}$$

$$\rho_{opponent} = \frac{\beta t}{(\beta t + 1)}. \tag{6}$$

From (1)–(6), it can be seen that strategy $\mathcal{T}$ evaluates and generates a set of issues according to an agent's deadline, the importance it attaches to an issue, its reservation values and the opposing agent's concession rates.

C. Network-Aware Strategy ($\mathcal{Q}$)

The time-dependent strategy, $\mathcal{T}$, can be modified into a network-aware strategy, $\mathcal{Q}$, in which an agent adapts its negotiation behavior to variations in the quality of the network (as discussed in Section III-B). Since our focus is on the influence of the underlying mobile environment, we extend only the time-dependent strategy to be network-aware, although more complex strategies that are independent of the network [21] could be similarly extended. From the list in Section III-B, we choose quality of service of the network as a parameter in our algorithms because QoS encapsulates many of the other features (in particular latency and bandwith variations [15]). Moreover, there are already substantial research efforts that relate QoS to the characteristics of mobile computing environments where bandwidth, throughput, timeliness, reliability, cost and perceived quality are the foundations of QoS [15].

In this context, $(x^t_{a\to b})[j]_\mathcal{T}$ denotes the response computed by agent *a* to agent *b* for issue j at time t using the above strategy $\mathcal{T}$. Similarly, $(x^t_{a\to b})[j]_\mathcal{Q}$ denotes the response computed by agent *a* to agent *b* for issue j at time t using the strategy $\mathcal{Q}$ below. Agent *a* adjusts its computed response $(x^t_{a\to b})[j]_\mathcal{T}$ according to strategy $\mathcal{Q}$ to obtain a new network-aware response $(x^t_{a\to b})[j]_\mathcal{Q}$. We take care that adjustments to obtain $(x^t_{a\to b})[j]_\mathcal{Q}$ do not yield a set of issues outside the reservation values of agent *a*. In addition, the new set of issues computed from strategy $\mathcal{Q}$, at time t, should not undo the concessions at time $(t-1)$ and previously. For example, if *a* decreased the price from $25 to $20 at time $(t-1)$, then the network-aware strategy at time t returns a price not more than $20. This is necessary because otherwise an agent would erratically concede and nullify its previous concessions with improvements in the network quality. This would, in turn, reverse the convergence toward an agreement between the parties. Given this constraint, two cases arise: 1) there is no previous message at $(t-1)$ i.e., $(x^t_{a\to b})[j]_\mathcal{Q}$ is the first message *a* is sending to *b* in that negotiation instance; 2) there is a previous message $(x^{t-1}_{a\to b})[j]$. Let Q embody the quality of the network between agents *a* and *b* and be parameterised from the bandwidth, latency, error rates and rate of disconnections (as discussed above). The response of *a* to *b* according to strategy $\mathcal{Q}$ is calculated as below. The final response for the network-aware behavior considers both $(x^t_{a\to b})[j]_\mathcal{T}$ and $(x^t_{a\to b})[j]_\mathcal{Q}$ depending on the criticality of the deadline versus the quality of the network.

1) No previous messages at $(t-1)$

$$(x^t_{a\to b})[j]_\mathcal{Q} = \frac{(Q \times (2 \times \max^a_j - \min^a_j)) + \min^a_j}{(Q+1)}, \quad \text{if } V^a_j \text{ is decreasing} \tag{7}$$

$$(x^t_{a\to b})[j]_\mathcal{Q} = \frac{(Q \times (2 \times \min^a_j - \max^a_j)) + \max^a_j}{(Q+1)}, \quad \text{if } V^a_j \text{ is increasing.} \tag{8}$$

2) Previous Message $(x^{t-1}_{a\to b})[j]$

$$(x^t_{a\to b})[j]_\mathcal{Q} = \frac{(Q \times (2 \times x^{t-1}_{a\to b})[j] - \min^a_j)) + \min^a_j}{(Q+1)}, \quad \text{if } V^a_j \text{ is decreasing} \tag{9}$$

$$(x^t_{a\to b})[j]_\mathcal{Q} = \frac{(Q \times (2 \times (x^{t-1}_{a\to b})[j] - \max^a_j)), +\max^a_j}{(Q+1)} \quad \text{if } V^a_j \text{ is increasing.} \tag{10}$$

Strategy $\mathcal{Q}$ exponentially adjusts the value of an issue as the QoS of the network varies. However it is ensured that the value of the issue is between its reservation values, but not less than what was sent by that agent before (to ensure consistent concessions). For example, if an agent prefers a high price, then with increasing quality of the network, the value of the issue that an agent generates tends to what it last sent or its maximum value. Similarly if the QoS tends to zero, then the value of the issue tends to its minimum acceptable value. Thus as the quality

TABLE III
HISTORY ABOUT OPPONENT FOR UNSUCCESSFUL NEGOTIATIONS

Criteria	Label	Weight
Times *Y*'s offer was not chosen	S_{r1}	w_1
Times *Y*'s offer was not cheapest in band	S_{r2}	w_2
Times *Y* rejected	S_{r3}	w_3

of the network degrades, the agent concedes more and a better quality of the network implies less tendency to concede.

D. Experience Strategy ($\mathcal{E}$)

The strategy $\mathcal{E}$ models the experience of an agent regarding negotiations in general and similar or parallel negotiations in particular. Thus, an agent may have experience about previous negotiations for the same items, with the same opponents, with similar preferences and environmental constraints. Such experience may also be gained from parallel concurrent negotiation threads in which an agent is engaged. For example, an agent *a* may know that negotiation with agent *b* usually requires 10 s or 20 messages or that *b* is unreliable. An agent *a* may also have some idea of the preferences of its opponents and this can be used to provide a faster convergence to an acceptable response. Added to network-awareness, agent *a* may also know the quality of network between itself and *b* from concurrent or recent negotiations or from information gathered from other neighbors or opponents of *b*. For example, agent *a* may know from past experience that the network connection between itself and agent *b* deteriorates over time and therefore *a* chooses to concede fast or not to bargain so as to reach a satisfactory agreement with a minimum number of messages exchanged.

In more detail, let E quantify the influence of an agent's experience on its decisions. The higher E, not only the more experienced is an agent, but also the more it chooses to apply its experience rather than ignore it (as in the case of no time or lack of computational power). The measure E is thus quantified according to how much experience an agent both has and uses for that particular negotiation instance. An agent's experience depends on a number of factors and below we list those that are relevant to automated service negotiation. These factors are used to compute an agent's experience, E, for the experience strategy ($\mathcal{E}$). The experience of agent *X* about dealings with an opponent *Y* is divided into three categories which broadly cover most of the features of previous negotiations with *Y*:

1) when negotiations with *Y* ended with a rejection;
2) when the opponent *Y* accepted offers;
3) quality of the deals with *Y*. These experiences and the corresponding weights attached to them are detailed in Tables III–V (for each category) and Table VI.

Note that an agent can vary the importance attached to the different aspects of previous negotiations by varying the associated weights. An item is associated to a band, for example, a low, medium or high price band.

TABLE IV
HISTORY ABOUT OPPONENT FOR SUCCESSFUL NEGOTIATIONS

Criteria	Label	Weight
Times *Y* said yes	S_{a1}	w_4
Times *Y* said yes in a particular band	S_{a2}	w_5
Times *Y* said yes for specific item	S_{a3}	w_6

TABLE V
QUALITY OF THE DEALS

Criteria	Label	Weight
Total number of agreements with *Y*	S_{o1}	w_7
Number of items obtained from *Y* for a particular band	S_{o2}	w_8
Number of specific items obtained from *Y*	S_{o3}	w_9

TABLE VI
INFORMATION ABOUT AN OPPONENT

Score	Label	Weight
Irrespective of item and band $(w_7 \times s_{o1}) + (w_4 \times s_{a1}) + (w_1 \times s_{r1})$	S_{T1}	w_{12}
Based on band $(w_8 \times s_{o2}) + (w_5 \times s_{a2}) + (w_2 \times s_{r2})$	S_{T2}	w_{11}
Based on Item Type $(w_9 \times s_{o3}) + (w_6 \times s_{a3}) + (w_3 \times s_{r3})$	S_{T3}	w_{10}
Based on Quality of Network	S_{T4}	w_{13}

Let *X* denote the agent whose experience is being modeled and, therefore, *X* records information about the history of its negotiations with opponent *Y*. Table III records the causes for previously unsuccessful negotiations with agent *Y*. For example, the first row labels S_{r1} as the number of times *Y* made an offer to *X* but *X* did not choose to agree. The weight associated to this criteria is w_1. Table IV records details of previously successful negotiations with agent *Y*. For example, row 2 in Table IV references the number of times (S_{a2}) that *Y* agreed in a particular band and weight w_5 is attached to it. Table V represents the quality of the deals obtained from previous negotiations with *Y*. For example, the number of deals for a particular item is shown in row 3 as (S_{o3}) and is given weight w_9. Table VI aggregates the various information on agent *Y*. Row 1 records positive responses from *Y*, row 2 the positive responses per band, row 3 the positive responses based on the type of the items and row 4 based on the quality of the network connection with *Y*.

The measure E is computed and normalized from S_{T1}, w_{12}, S_{T2}, w_{11}, S_{T3}, w_{10}, S_{T4} and w_{13}. For example, agent *X* modeling negotiations with agent *Y* calculates E as follows:

$$E = f_X(S_{T1} \times w_{12} + S_{T2} \times w_{11} + S_{T3} \times w_{10} + S_{T4} \times w_{13}) \quad \text{where } E \to [0,1]. \qquad (11)$$

The set of issues $(x^t_{a\to b})_{\mathcal{E}}$ (using the experience strategy) is calculated in a similar way to the network-aware issue set $(x^t_{a\to b})_{\mathcal{Q}}$, but instead substituting Q by E. For example for the case of decreasing V^a_j and *a*'s previous message being $(x^{t-1}_{a\to b})$ the issue j that *a* generates at time t using the experience strategy, $\mathcal{E}$, is calculated as

$$(x^t_{a\to b})[j]_{\mathcal{E}} = \frac{((E \times (2 \times (x^{t-1}_{a\to b})[j]) - \min^a_j) + \min^a_j)}{(E+1)}. \qquad (12)$$

Therefore, the above strategy $\mathcal{E}$ considers previous negotiations with a particular agent in order to guide its responses in current negotiations with the same opponent. The experience strategy, $\mathcal{E}$, may be combined with the network aware strategy, $\mathcal{Q}$, for a hybrid strategy to behave flexibly to more than one operational factor. For example, the hybrid strategy $\mathcal{E}$ and $\mathcal{Q}$ could allow an agent to infer that given that the quality of communication is currently poor (from the network-aware strategy) and it has been so for the last 2 h (from the experience strategy), then it will most likely remain so for the next 10 min. This would be especially useful if the agent's deadline lies before the next 10 min.

TABLE VII
BILATERAL ALGORITHM FOR ACCEPTABLE ISSUE SET ($\mathcal{AC}$)

```
if $((t+1) > t^a_{max})$ then
   $next\text{-}action_a$ = **timeout** ; **exit;**
if not $(V^a(x^t_{b\to a})$ *lies within* $D_a)$ then
   sub-procedure algorithm $\mathcal{AC}$ in Table VIII; **exit;**
else
  if (*state* == **offered(b)** ∧ ¬**proposed(b)**)
        $\vee((no\text{-}time\text{-}left) \vee better\text{-}than \wedge$
         (**a.agree** $\in set\text{-}of\text{-}possible\text{-}next\text{-}actions)$ then
     $next\text{-}action_a$ = **a.agree** with $\mathrm{x}^t_{b\to a}$
  elseif $(no\text{-}time\text{-}left \vee better\text{-}than)$
    $next\text{-}action_a$ = **a.offer** $\mathrm{x}^t_{b\to a}$
  elseif $(distance == close \vee middle) \vee less\text{-}time\text{-}left$
     if (**a.propose** $\in set\text{-}of\text{-}possible\text{-}next\text{-}actions)$ then
        $next\text{-}action_a$ = **a.propose** $\mathrm{x}^{t+1}_{a\to b}$
    else $next\text{-}action_a$ = **a.counter_propose** $\mathrm{x}^{t+1}_{a\to b}$
   elseif $(distance == far) \wedge (middle\text{-}time\text{-}left \vee much\text{-}time\text{-}left)$
     if (**a.suggest** ∈ set-$of\text{-}possible\text{-}next\text{-}actions)$ then
        $next\text{-}action_a$ = **a.suggest** $\mathrm{x}^{t+1}_{a\to b}$
    else $next\text{-}action_a$ = **a.request** $\mathrm{x}^{t+1}_{a\to b}$
  endif
```

TABLE VIII
ALGORITHM FOR UNACCEPTABLE ISSUE SET ($\mathcal{NAC}$)

```
if $state == (offered(b) \wedge \neg proposed(b))$ then
   $next\text{-}action_a$ = **reject** $x^t_{b\to a}$ ; **exit;**
elseif $no\text{-}time\text{-}left$ then
   if (**a.offer** $\in set\text{-}of\text{-}possible\text{-}next\text{-}actions)$ then
     $next\text{-}action_a$ = **a.offer** $\mathrm{x}^{t+1}_{a\to b}$
   else $next\text{-}action_a$ = **a.propose** $\mathrm{x}^{t+1}_{a\to b}$
  elseif $(distance == close) \vee less\text{-}time\text{-}left$ then
     if **a.propose** $\in set\text{-}of\text{-}possible\text{-}next\text{-}actions$ then
        $next\text{-}action_a$ = **a.propose** $\mathrm{x}^{t+1}_{a\to b}$
    else $next\text{-}action_a$ = **a.counter_propose** $\mathrm{x}^{t+1}_{a\to b}$
   elseif $(distance == middle \vee far) \wedge$
             $(middle\text{-}time\text{-}left \vee much\text{-}time\text{-}left)$
     if **a.suggest** ∈ set-$of\text{-}possible\text{-}next\text{-}actions$ then
        $next\text{-}action_a$ = **a.suggest** $\mathrm{x}^{t+1}_{a\to b}$
    else $next\text{-}action_a$ = **a.request** $\mathrm{x}^{t+1}_{a\to b}$
  endif
```

V. COMBINING THE PROTOCOL AND THE STRATEGIES

The two negotiation algorithms $\mathcal{AC}$ and $\mathcal{NAC}$ in, respectively, Tables VII and VIII combine the bilateral protocol of Fig. 4 with either of the three strategies $\mathcal{T}$, $\mathcal{Q}$ and $\mathcal{E}$ (in Section IV). These algorithms specify the decisions by agent *a* with deadline $t^a_{\max}$, on receiving a set of issues $x^t_{b\to a}$ and a state *state* (e.g., *offered* or *requested*) from agent *b* at time t. Agent *a* responds with $x^{t+1}_{a\to b}$ if needed. Let the flag *better-than* be set to $V^a(x^t_{b\to a}) \geq V^a(x^{t+1}_{a\to b})$ (i.e., agent *a* has received a response from *b* better than it would have sent). A set of issues received from *b* is compared with *a*'s goals to analyze the difference (given by *distance*) between what *a* received at time t and what it will send next at time $t+1$ (i.e., the variable *distance* measures, with respect to the score of the issue set received and sent, how far *a* and *b* are to an agreement). Here *distance* has delimiters *close, middle* and *far*, ordered increasingly with the difference of the valuation of *a* between its own and *b*'s responses ($|V^a(x^t_{b\to a}) - V^a(x^{t+1}_{a\to b})|$). The domain (preferences) of *a* is denoted by D^a. The set of possible next actions, *set-of-possible-next-actions*, can be derived from the bilateral protocol and $next - action_a$ denotes the decision for the next state-triggering action by *a*.

In algorithm $\mathcal{AC}$, *a* sends a *timeout* if its deadline has expired. Otherwise, if $V^a(x^t_{b\to a})$ is not acceptable then *a* follows algorithm $\mathcal{NAC}$. A set of issues is acceptable if each issue lies within the reservation values or in the qualitative set of *a*. The rest of algorithm $\mathcal{AC}$ considers the decisions of *a* after it has received an acceptable set of issues. If the state is *offered(b)* and not *proposed(b)* (take it or leave it acceptable offer from *b*), then *aagrees*. If $V^a(x^t_{b\to a})$ is better than what *a* would have sent (*better-than* holds) or if *a*'s deadline is close, then *a* sends an agree, if the state is *offered,* else an offer. If *a* does not have much time left or *better-than* holds, then *a* does not bargain and sends an offer with what *b* sent it. However if *a*'s deadline is not close, then *a* makes a proposal if they are close or middle to an agreement. In so doing, *a* moves to a higher level of commitment than a *suggest* so as to reach an agreement faster. Otherwise, if an agreement is not within reach and there is enough time left, then *a* bargains with *suggest* and *request.*

Algorithm $\mathcal{NAC}$ portrays the decisions of *a*, at time $t+1$, when it receives an unacceptable set of issues from *b* at time t. In this algorithm, *a* does not find it worthwhile to agree to $x^t_{b\to a}$ and responds with a more favorable set of issues, $x^{t+1}_{a\to b}$. If *a*'s deadline is close, then *a* refrains from bargaining and sends an offer or proposal with $x^{t+1}_{a\to b}$, as a penultimate step to termination. Otherwise if an agreement is close, then *a* triggers the *proposed* state. If an agreement is not near and there is enough time, then *a* bargains through requests and suggestions.

The characteristics of the network or an agent's experience are used when evaluating the set of issues $x^t_{b\to a}$ and generating $x^{t+1}_{a\to b}$, which are then used in algorithms $\mathcal{AC}$ and $\mathcal{NAC}$ and in the conditions *better_than* and *distance*. An agent can flexibly choose its strategies privately and reuse the above two algorithms given that the protocol is public and complied with. The calculated set of issues from other strategies can be adjusted to take into account the m-service's domain through the eventual combination with strategy $\mathcal{Q}$. Thus, in our approach, it is easy to adjust the response of an agent if it chooses to consider the variations in the quality of the network. As a result, our algorithms $\mathcal{AC}$ and $\mathcal{NAC}$ and the three specified strategies can be combined with other strategies so that the latter strategies become adaptive to time-constraints, communication capabilities or the social environment.

VI. PERFORMANCE ANALYSIS OF THE NEGOTIATION MECHANISMS

This section is an initial evaluation of our m-commerce negotiation mechanisms. It concentrates on varying the adaptative capability of the agents in relation to the quality of the network and their negotiation experience. More specifically, the performance of the algorithms is analyzed with respect to a set of identified performance metrics for automated negotiation. The task of such a performance analysis is to explain the run-time behavior of the system configurations, compare between strategies, and ultimately to validate and optimize our algorithms.

There are a number of parameters that affect the performance of an m-service negotiation. However, in conducting our experiments, we choose to vary those that we believe are the most obvious in this domain. These include the following.

- *Various characteristics of mobile telecommunication networks* (including QoS, latency and error rates as discussed in Section III-B). These are used in strategy $\mathcal{Q}$ when calculating the rate of concessions.
- *Adaptability of an agent*. This depends on which strategies an agent chooses and range from no negotiation at all, to complex calculations about experience, observed behaviors and quality of service. The strategies in Section IV are varied and combined to analyze their performance relative to each other. We are then able to analyze the benefits of adapting to the environment and the features in Section III-B.
- *Complexity of the tasks and the agents' workloads*. These are varied by altering the number of issues in a negotiation, the dependency between the issues and the difference between the agents' preferences. In particular, we explore how varying the initial preferences and the number of issues influence the quality of the deals reached.
- *Resources of the participants*; including their deadlines, money and reservation values (as discussed in Section III-C). The deadlines and reservation values of the agents are varied to see how the negotiation degrades as the quality of the network decreases or the deadlines becomes closer.
- *Knowledge and beliefs of an agent about its opponents*. This includes experience from past negotiations for prediction about future ones and trust of other agents. The experience of the agents is captured through E (as in the strategy $\mathcal{E}$) and varied versus the quality of the deal reached and the quality of the network.
- *Number of concurrent threads per agent*. In this paper, we choose thirty concurrent negotiations between buyer and seller agents, with the set of issues consisting of price, delivery time and quantity.

As we vary the above parameters, we analyze the adaptive behavior of the agents through the following performance metrics.

- *Whether an agreement has been reached*. Thus we analyze whether the negotiating parties manage to converge to an agreement.
- *Values of the issues and the utility of the exchanged set of issues over time*. The utility of an exchanged issue set is computed by the receiver agent using the evaluation function defined in Section IV-B. These values and utilities are measured for each negotiation thread and include the utility of the deals that are agreed upon.
- *Rates of concessions of the buyer and seller agents*. These can be calculated from the difference between consecutive sets of issues sent by an agent divided by the time period over which they are sent ($|V^a(x^{t'}_{b\to a}) - V^a(x^{t}_{b\to a})/t' - t|$).
- *Times taken* to evaluate, generate, send and receive messages, and time for agreements to be produced.
- *Processing costs of negotiation*. These include the addition of the costs for evaluating and responding to a received set of issues and for message exchange.

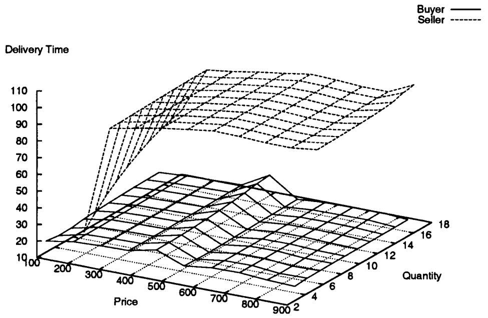

Fig. 5. Thirty concurrent buyer–seller negotiations.

- *Number of message exchanges per negotiation*. This is simply the sum of the messages sent by both agents.

The remainder of this section presents an evaluation, with respect to the above metrics, of the strategies identified in Section IV, while varying the above parameters. In the simulations, buyer and seller agents conduct multiple concurrent negotiations with each other by exchanging a set of issues over an item. Let s denote the seller and b denote the buyer. The items are separated into price bands low, medium and high. The issues being negotiated are price, delivery and quantity. We choose these issues because they occur frequently in our motivating scenarios and they are reasonably generic. Each agent has an initial preference over the issue set where buyer agents are designed to prefer low prices, short delivery times and low quantity. Sellers are designed to prefer high prices, long delivery times and high quantity. Each agent has its own (private) deadline to conclude a particular negotiation. Finally, seller agents start the negotiation by sending a message according to their initial preferences and resources. Let s denote the seller and b denote the buyer.

The behavior of the negotiation model is evaluated by conducting a series of experiments. First, in Section VI-A, we investigate if the agents manage to converge or find agreements as their deadlines vary or when an agent has a very close deadline. This is to analyze the agents' behaviors given different constraints on time. Second, in Section VI-B, we explore how the agents vary the set of issues they exchange as their available resources change over a negotiation instance. We measure the utility of the exchanged set and the deals reached and we compare the agents' behavior with respect to different negotiation threads. In this context, different negotiation threads show different agents' behaviors when their preferences varies. Therefore to compare the resources used for different levels of interactions, third, in Section VI-C we analyze the processing costs for a negotiation that consists only of an offer-agree interaction and for more interactive forms of negotiation. Naturally, we expect the costs of negotiation to be more than the fixed price model, but we also expect some benefit from negotiation in terms of obtaining information about the participants (which in essence is part of the experience strategy). In addition, for evaluating the experience strategy, we conduct concurrent negotiations so as to both reflect realistic situations and to observe how the experience of parallel negotiation instances influence an agent's behavior.

A. Reducing Negotiation to a Single Offer

An alternating offers bargaining model is used in [22] and [23] for computationally limited negotiations. It is shown that the equilibrium strategies for the model result in a single shot take-it or leave-it strategy. Thus the agents wait without exchanging offers until one of their deadlines arrives and then the agent with the earlier deadline concedes and makes an offer that the other agent may accept. Fig. 5 reflects this behavior for the agents with closer deadlines. The top (dotted) graph represents the final issue set obtained by the seller agent and the lower one the final issue set obtained by the buyer agent. In this experiment, the deadline of at least one of the parties, the seller here, $t^s_{\max}$, is varied so as to lie close to the start of the negotiation. This means that one party has significantly less resources than the other and, therefore, is at a disadvantage or both parties have close deadlines. In this way, we can analyze the behavior of the agents when they have no time left, *no_time_left* is true, or there is a disparity in their relative amount of resources available. Specifically, we investigate if the $\mathcal{T}$ strategy allows the agents to still be able to find a deal. Here, we observe that they do so by one party, the one with the closer deadline, making a large concession and basically sending a take-it-leave-it offer that precludes any bargaining. If both agents have close deadlines, the first agent that initiates the negotiation (the seller in this case) makes a large concession.

Reducing the bilateral negotiation mechanism to a *single* offer-accept or reject strategy, without prior suggestions and iterations, has the advantage of decreasing the costs by sending fewer messages and allowing agreements to be found without delay. However assuming that deadlines are common knowledge is not practical, as found in [6]. We do not assume this here. Moreover a single shot negotiation does not embody bargaining. Normally the two agents have different preferences and their intersection is not guaranteed since the agents have no sure information about their opponents' strategies and preferences. In such cases, negotiation allows agents to probe each other's range of acceptability so that they may move their demands and issue set to create a mutual acceptance region. This process could take the form of a "trial and error" exchange of messages where the agents are providing feedback and learning what is acceptable. Given this, it is hard to see how a single offer negotiation allows the agents to revise their beliefs and preferences to converge toward an agreement. Moreover, proactive agents that negotiate in advance of their deadlines, so as to plan ahead or schedule their tasks, are discouraged by a reactive strategy. Also, in parallel negotiations, a seller or buyer may find other interesting deals rather than wait for a deadline to arrive.

In summary, the aim of this simulation is to observe how the agents adapt when they have close deadlines. This relates to the feature in Section III-C that negotiations in the m-commerce domain usually have deadlines. Some of these deadline may become close because of repeated disconnections, increased latency or deterioration of the network due to a change in location. From this set of experiments, we can conclude that the time-aware strategy $\mathcal{T}$ ensures that agents can still find agreements in such time-constrained environments. It does so by conceding and making a final offer that is acceptable to its opponent.

B. Valuation of Deals for Concurrent Negotiations

This section focuses on the exchanged set of issues and their valuation within three concurrent negotiation threads between buyer and seller agents. Table IX gives the minimum and maximum prices of the seller ($[\min^s_{price}, \max^s_{\text{price}}]$) and buyer ($[\min^b_{\text{price}}, \max^b_{\text{price}}]$) agents for low, medium and high item bands. We choose three threads to study the agents' behavior for each band.

TABLE IX
RESERVATION VALUES FOR PRICE FOR BUYERS AND SELLERS

Agent	Band	Minimum	Maximum
Buyer	Low	100	200
Buyer	Medium	400	600
Buyer	High	800	900
Seller	Low	80	180
Seller	Medium	380	580
Seller	High	780	880

TABLE X
INITIAL PREFERENCES OF BUYERS AND SELLERS

Agent	Thread ID.	Band	Price	Quantity	Delivery
Buyer	1	Low	109	6	131
Buyer	2	Medium	562	6	105
Buyer	3	High	805	5	81
Seller	1	Low	100-200	1-20	5-120
Seller	2	Medium	400-600	1-20	5-120
Seller	3	High	800-900	1-20	5-120

TABLE XI
WEIGHTS OF ISSUES BY AGENTS

Agent	Price	Quantity	Delivery
Buyer	0.3125	0.625	0.0625
Seller	0.286	0.571	0.143

Table X shows the initial preferences of the buyer and seller for each of their negotiation threads. For example, the first row of Table X shows that in the first negotiation thread, the buyer agent prefers a low band item, with price 109, quantity 6, and delivery time 131. Similarly the 6th (last) row shows that in its third negotiation thread, the seller prefers a high band item, price between 800 and 900, quantity 1–20 and delivery time between 5–120. Table XI indicates the weights each agent associates with the various negotiation issues. These weights are the same for a given agent, for all its concurrent negotiation threads. Moreover, an agent associates the same importance to an issue in all its concurrent negotiations. Specifically, the weights w_1 to w_{12} in Table VI, used in the experience strategy, for the buyer are, (w_1, 11), $(w_2, 5), (w_3, -1), (w_4, 20), (w_5, 5), (w_6, -1), (w_7, 20)$, ($w_8$, 5), ($w_9, -1$), ($w_{10}$, 20), ($w_{11}$, 5), ($w_1$2, 1) and for the seller are (w_1, 20), (w_2, 5), (w_3, 20), (w_4, 5), (w_5, 20), (w_6, 5), (w_7, 20), (w_8, 5), (w_9, 20), (w_{10}, 10), (w_{11}, 2), ($w_1$2, 20).

Fig. 6 shows the values of the issues in the set received by the buyer agent ($x^{t_n}_{s\to b}$) during its concurrent negotiation threads (Buyer Thread 1, Thread 2 and Thread 3). The axes denote the issues price, quantity and delivery. Recall the seller prefers high prices and high delivery times. As can be seen, each negotiation thread produces a different surface in the graph corresponding to the band of the item being traded; thread 1 has a low band price, thread 2 a medium band price and thread 3 a high band (as given in Table X).

In Fig. 6, on all three surfaces, there is one point with a high delivery time that stands out. This point shows the seller starting the negotiation (say at time t_0) by sending a set of issues ($x^{t_0}_{s\to b}$) near to its preferences (since it has no knowledge of the initial

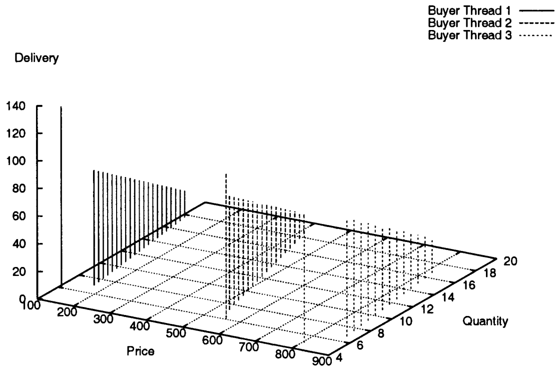

Fig. 6. Buyer's issue-value dynamics received from the seller for three negotiation threads.

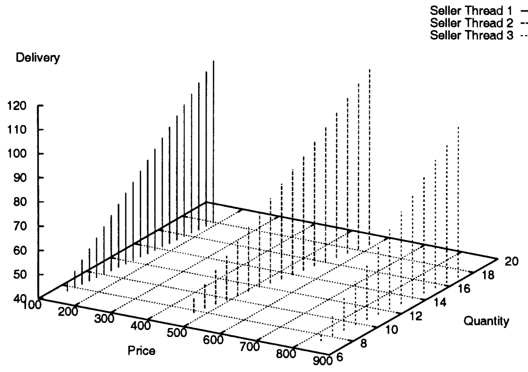

Fig. 7. Seller issue-value dynamics for three negotiation threads.

preferences of the buyer). When the seller receives the first issue set from the buyer ($x^{t_1}_{b\rightarrow s}$), it updates its issue set to reflect the knowledge it has just learnt about the buyer. This explains the large difference, on each of the surfaces, between the first and second issue set ($x^{t_2}_{s\rightarrow b}$) received by the buyer.

After sending the second issue set, the seller consistently concedes. It decreases the price at a lower rate than both decreasing the quantity and increasing the quanity. These consistent changes in values lead to an overall concession by the seller, when multiplied by the relative weights of each issue [as per (1)]. The same behavior regarding the seller's concessions (from evaluating the messages received by the buyer) is reflected for all three negotiation threads and the corresponding three surfaces in Fig. 6. It should also be noted that for the high band item, fewer messages are received by the buyer, than for the medium priced item and even fewer for the low priced item. For a high-priced item, the agents perform more computation before sending a message (because they risk more money), resulting in fewer messages being sent in that negotiation.

In a similar vein, Fig. 7 shows the seller's version of the values of the issue set it received ($x^{t_n}_{b\rightarrow s}$) during the three concurrent negotiation threads (Seller Thread 1, Thread 2, and Thread 3). Again there are three distinct surfaces to reflect the fact that the item in each negotiation thread has a different band price. Since the buyer does not start the negotiation, it consistently concedes toward the seller's preferences. Thus, there is no initial point uncommonly near to the buyer's preferences at the start. As a negotiation progresses, the buyer responds to the seller's message by asking for less quantity without proportionally decreasing the price, but increasing its delivery time to the buyer's advantage. Overall, the buyer moves toward the seller's preferences by "conceding" in offering less price, but with less quantity. This is offset by asking for a lower delivery time, since the buyer places more relative importance on delivery times and less on price and quantity than the seller. In addition, Fig. 7 shows, for each thread, the number of messages received by the buyer matches the number received by the seller in Fig. 6 (as would be expected given the alternating nature of the protocol).

Fig. 8 aggregates the two graphs, in Figs. 6 and 7, and is taken from a particular angle. The graph shows six threads – three concurrent negotiations run by each agent. The surfaces in the graph are grouped in pairs with Thread x ($1 \leq x \leq 3$) for a buyer agent coupled with Thread x for a seller agent. Thus each pair shows the message exchange between these two threads. As in Fig. 6, the graphs show the initial set of issues received ($x^{t_0}_{s\rightarrow b}$) and evaluated by the buyer to be especially close to the seller's preferences. After the seller's first message, each pair of surfaces show the convergence of the two surfaces toward each other. This means that the buyer and seller are conceding according to the time they have left, and they are therefore showing time-aware behaviors. Each pair also shows a similar number of messages being exchanged per agent and negotiation instance, and indicates the total amount of message exchange per negotiation instance.

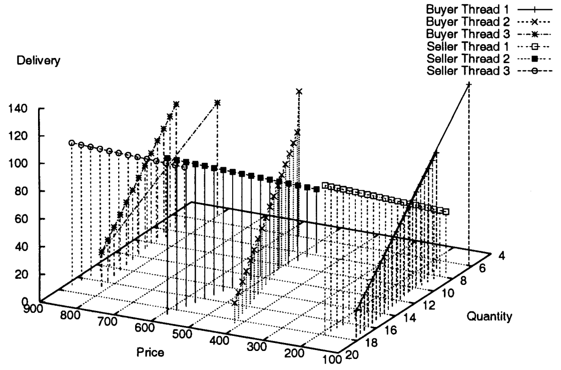

Fig. 8. Buyer and seller issue-value dynamics for three negotiation threads.

Fig. 9 shows the scores of the points in Fig. 8 along time. It expresses the score of the issue set received and evaluated by each agent versus time, for each negotiation thread. The lines labeled with Buyer Thread 1 to Thread 3 show, for each thread, the sequence of the buyer's evaluations of the messages it receives from the sender as the negotiation progresses (*mutatis mutandis* for seller Thread 1 to Thread 3). For the same negotiation instance (e.g., seller's Thread_x and buyer's Thread_x), an agreement is reached with a particular issue set, say issue set z. Each agent obtains a different score for this agreement, according to their chosen weights for an issue and their utility function ($V^s(z)$ or $V^b(z)$). Similarly to Fig. 8, the lines labeled Buyer Thread_x and Seller Thread_x are coupled, since they are exchanges regarding the same negotiation instance, about the same item. The band price determines the region in which the a pair of lines lie.

The lines labeled Buyer Thread_x demonstrate that as time elapses, the buyer believes the seller is conceding and gaining

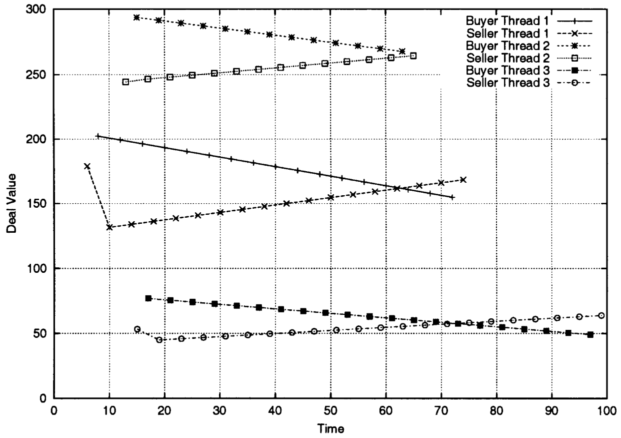

Fig. 9. Buyer and seller deal value dynamics for three negotiation threads.

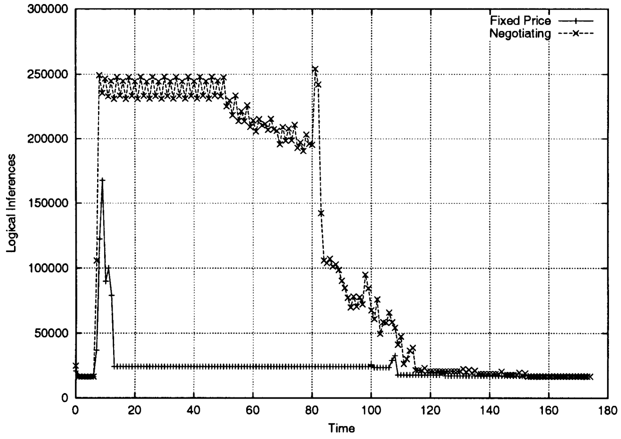

Fig. 10. Processing costs when negotiating and not negotiating.

less. Although the gradient of the three Buyer Thread_x lines are negative, the rate of concessions by the seller differ for each negotiation thread. We observe that the line labeled with Buyer Thread 3 is less steep than that for Buyer Thread 2. This means that in time-constrained environments, *the seller adapts by conceding more for a low band and less for a higher band* because the seller is selling more low band items than high band ones. Therefore the seller is ready to concede more for a lower band item so as to avoid a rejection.

The lines labeled with Seller Thread_x shows the seller's evaluation of the issue set it receives from the buyer ($V^s(x^{t_n}_{b\rightarrow s})$). According to the seller, the buyer gains over time during a negotiation (given by a positive gradient). Similarly for the seller threads, it can be shown that the gradient of the lines are less steep for a higher price bands than lower price bands. Thus in time-constrained negotiations, contrary to the seller, *the buyer adapts by conceding more for higher band prices* because it buying less higher band items and therefore can afford to concede more, depending on the total price.

It should also be noted that each line in a pair, expressing each agent's response in an instance, converge toward an intersection. This shows the score for each buyer and seller moves toward an agreed set of issues. This agreed set yields a different score for each participant, depending on their preferences.

Fig. 9 also shows that it is not necessarily the least expensive items that produce a higher score. For example, Buyer and Seller Threads 2 produce a higher score than Threads 1 and Threads 3. We can thus conclude that *it is more profitable for both the buyer and seller to negotiate about medium band items, than either high or low band items* because Figs. 6 and 7 show that they both obtain more quantity for medium band items than in the other two bands. In addition, according to the weights in Table XI, they both attach more importance on the quantity of items than the price. Hence by (1), they achieve more for medium band items. In turn, *it is more profitable for them to negotiate about low band items than high band items* because the quantity of items in the final deal is higher for a low band item than for a high band one. It makes sense that the buyer cannot afford to buy high-priced items.

In summary, in this experiment, we consider how the strategies for time $\mathcal{T}$ and experience-awareness $\mathcal{E}$ allow an agent to adapt given it has different negotiation instances and different preferences. The resources that vary in this domain include time, the price an agent can afford, how soon does it needs the item (delivery time) and the desired quantity of the item. As these resources vary, we see that the agents concede accordingly, depending on the importance they place on each issue. Having no experience experience, the first message sent by an agent often lies outside the other agent's acceptable region. Therefore experience-awareness helps to start the negotiation with an acceptable issue set, especially in situations where the agents have low resources even at the start. Furthermore, experience awareness allows the initiator to conceal its preferences from its opponents since the initiator does not start the negotiation with an issue set close to its own preferences. We also conclude that in m-commerce environments (where timeouts, disconnections, the time for message exchange, and latency affect the time an agent has to find an agreement) then by using our strategies, agents can better adapt to the time-constraints to find agreements.

C. Processing Costs for Negotiation

There are two types of interactions that can take place in this simulation – with and without negotiation. When not negotiating, the buyer makes a one-shot offer to the seller (as per Section VI-A). The seller then either accepts or rejects it. Fig. 10 shows the processing costs incurred by agents undertaking 30 simultaneous purchase/provision processes (i.e., they are each buying or selling 30 items in parallel). As above, the buyer and seller are given the same preferences and reservation values. Buyer agents retain preference for low price and quantity and short delivery times, and sellers prefer high prices and quantities with long delivery times. In Fig. 10, the vertical axis (*Logical Inferences*) represents resource utilization by agents- including both processing (evaluation and response generation) and message exchange. The lower line shows the total costs for all seller and buyer agents utilizing fixed pricing, while the higher line shows the costs for the negotiating agents (using the experience strategy $\mathcal{E}$).

With fixed pricing, the agents reach deals as soon as possible (with retry on rejection). This is respresented by the sharp spike on the left hand side. The agents then continue to incur some resource costs whilst they monitor for the correct delivery of items (buyers) or payment (sellers). All payments and deliveries are concluded by time 110. In comparison, the resource utilization of the negotiating agents is spread throughout time until all outstanding deliveries/payments are closed (at time 152). (Delivery times with the negotiating agents cover a wider range than their fixed price counterpart because this issue is negotiable). As can be seen, resource usage decreases with time because the number of open negotiations decrease as agreements are reached. Therefore there is a steady decline as deals are finalised and items are delivered or paid for.

In comparing the operational costs of negotiating and nonnegotiating agents, it is evident that negotiation is more costly. For this implementation, comparing the processing cost of agent, in total (i.e., including all other costs associated with the running agent) yields the following: a negotiating agent uses 4.96 times more resources than its fixed price counter-part. If the base operating cost of the agent (i.e., without resource costs related to other agent functionality) is removed from this calculation, negotiation is 14.8 times more costly than the fixed-price method. This value is high because negotiating agents retain state information that is necessary to enact their strategies (fixed price retains negligible state information). This includes information about the exchanged values during a negotiation (for purposes of examining concession rates, obtaining feedback for adaptation, etc.) as well as reasoning processes for ensuring coherence with the negotiation protocol. It is critical to be mindful that these specific operating figures cannot be assumed to hold beyond the context of the software used for this simulation. Although we believe that the broad trend is generic.

However, these results do not lead us to conclude that negotiation is bad. First negotiations allow the agents to change the set of issues so as to fall within the acceptable region of both parties. This flexibility is lost in fixed price trading since most of the time the agents do not know their opponents' preferences and utilities which often leads to rejections. Such rejections are a drain on the system's resources. In our negotiations, the agents perform inferences so as to store the state, the history of a negotiation, and the information about their opponents, the environment and how they themselves adapted to a resource-bounded environment. Thus, the agents accumulate experience. This experience is especially useful when using our experience-aware strategies since they allow adaptation to specific m-commerce environments. For example, knowing its opponent preferences and rates of concessions, an agent no longer needs to make an overly large concession if its deadline is very close. Rather, it only needs to make a concession that is just enough to obtain a deal. Also for agents initiating a negotiation, where these agents have a close deadline or their underlying connection limits their communication capability, then knowing what is likely to be a successful deal is crucial.

Second, although extra facilities and costs are required to support the strategies presented in this paper, (when compared to the costs for not negotiating), these facilities are likely to be required by other functional aspects of intelligent agents. Thus, much of the additional cost of negotiating using our strategies may be shared with other adaptive mechanisms in agents. For example, Fig. 11 shows a buyer agent's experience of negotiating with two suppliers. It portrays the final value of each deal ($V^b(z)$) against the number of message exchanges required versus the initial distance between the first issue/value set exchanged between the customer and supplier ($|V^b(x^{t_0}_{s\to b}) - V^b(x^{t_1}_{b\to s})|$). As can be seen, supplier 2 requires longer exchanges than supplier 1 and also results in a lower quality deal. Intuitively, it would be sensible for the customer to approach supplier 1 instead of 2. Yet, this does not take into account other operational factors that are important for agents in mobile domains. For example, the cost of communication (time, financial cost, etc) may differ according to the recipient and the agent's local context. Secondly, the need for predictability may be paramount. Thus, it may be necessary for an agent to be able to plan more effectively by reasoning more accurately about future expected resource costs. This would allow (and is required for) agents to actively manage risk [24]. (A risk averse agent may nevertheless choose to select supplier 2 because the region defining the costs associated with negotiating is less spread out than for supplier 1.Therefore it may be more certain about the costs it will incur during negotiation.) The agent may therefore trade utility for predictability thereby allowing it to plan its future resource commitments more effectively.

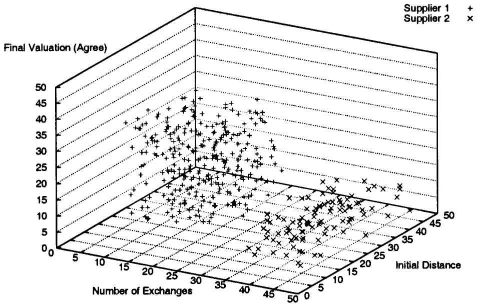

Fig. 11. Successful past negotiations.

A third advantage of negotiation in this context relates to the time taken for fixed price exchanges versus negotiation. In the example above, the fixed price mechanism is assumed to occur as far in advance of delivery time as possible. In this situation the customer and supplier agents must incur additional processing costs where nonpayment or nondelivery is possible. Because there is no contact between the customer and supplier between the time when a deal is reached and the delivery/payment time, there is an increasing degree of uncertainty present if the agent is operating in a domain where failure to deliver is an acceptable reality. At the other extreme, an agent that undergoes commodity/service acquisition just-in-time runs the risk of not reaching a deal in time. In contrast, the negotiating agent is in contact with the dialogue partner for a manipulable period (by managing time-outs and increasing negotiation time length by modifying concession rates). When combined with levels of commitment associated with different speech-acts of the protocol [24], both agents are able to manage uncertainty by having recently been in contact and also being able to infer information about the commitment to a deal of the negotiation partner.

In summary, this simulation evaluates the various costs of negotiation and compares them with a fixed price offer-agree interaction. We place our evaluation in the context of the experience strategy. Using such a strategy, an agent can adapt to a change of resources or limited resources as is frequently the case in m-commerce. Our model ensures an agent does not need to concede more than it should if it is running out of resources.

In addition, a seller can hide its true valuation or spend less resources on negotiation by sending an issue set nearer to a known customer's preferences. Finally, the experience strategy also allows an agent to accumulate information about its network connection with its opponent. Thus an agent may choose not to deal with a specific opponent because it knows that there is a faulty network connection between them which often results in the breakdown of their interactions. Moreover given concurrent negotiations, an agent may restrict the number of parallel threads it operates depending on the bandwidth limitations and fluctuations (as discussed in Section III-B).

VII. Related Work

Much of the related work regarding m-services is concerned with service discovery. For example, [25] provides an evaluation of the Jini discovery infrastructure and develops a framework for dynamic service discovery through a hybrid of service-oriented and agent-oriented architectures. Other work on service discovery includes [26] which surveys exisiting service discovery architectures, [27] which implements location aware agents embedded in mobile devices and seeking information via ad-hoc networks and teams, and [9] that presents an architecture for m-service discovery and brokering. Our work considers the next step on from m-service discovery, namely m-service negotiation.

There is substantial research on automated negotiation [4], [7], [21], [28], but none of this work specifically considers the m-service domain and the particular set of requirements it induces. As such, there are many different forms of negotiation (including auctions and bi-lateral encounters) that are tackled using many different techniques (including game theory, heuristics and argumentation) (see [4] for an overview). Here, however, we focus exclusively on bilateral encounters (since we found these to be especially prevalent in this domain [5]).

Some approaches to automated negotiation assume perfect rationality of agents, where the strategies and the best actions are computed instantaneously [7]. Although these lead to important theoretical contributions to negotiation strategies (such as optimality and equilibrium in constrained environments), the underlying assumptions are often inappropriate for practical contexts. Thus, our work seeks to develop models that can be used in practice (the downside of which is that results tend to concentrate on typical performance with few guarantees and outcomes need to be determined empirically). Therefore, we concentrate on ways to implement m-service negotiation and obtain empirical results. We do not assume that participants adopt a single dominant strategy because we are interested in how to design the various strategies that allow participants to bargain. Also, current research in AI negotiation and agent interaction tends to separate the design of intelligent strategies and the specification of agent communication languages and protocols. Such work tends to focus on either one or the other. In constrast, we seek to bridge this gap with a richer bilateral protocol, associated with resource-bounded strategies.

There is also significant research carried out in AI regarding dominant strategies, Nash equilibria, Bayes-Nash equilibria, Bayesian equilibria for auctions, load balancing or resource sharing bi-lateral negotiations. It would be useful to exploit such work for a theoretical evaluation of our different environmental strategies. However, there remain a number of open issues regarding with such work (including the generation of appropriate utility functions [24]). It is also notable that the agents in our scenarios are not negotiating for sharing a resource [29] or performing a joint task such as delivering parcels [23]. Although we could say the trading of an m-service is a joint task of exchanging the service, the competition is not in accessing the service but is rather in its profitable purchase and sale. Therefore, we adopt the intuitive service-oriented approach and notation in [20] and [21], which embodies the concept of electronic transactions.

Finally, most of the existing work on bargaining strategies applies to a single negotiation process or computing a single response. Generally speaking, it does not consider the possibility of concurrent negotiation threads. This is a shortcoming because the ability to exploit concurrent negotiation threads allows an agent to use the experience-strategy to achieve better deals (as discussed in Section VI-C).

VIII. Conclusions and Further Work

This paper proposes m-service negotiation as a key enabling technology for electronic commerce via agents located on mobile devices. In this domain the characteristics of wireless communications constrain interactions between agents and, therefore, this paper we identify and discuss those characteristics that most affect automated negotiation. Given these constraints, novel protocols and strategies for peer-to-peer negotiation between two agents are developed. In particular, three strategies are proposed that enable agents to adapt to their deadlines and preferences, to the quality of the network and to their experience regarding similar negotiations. These negotiation mechanisms are then analyzed and we can conclude that they allow negotiating agents to adapt to time and resource limitations in order to find agreements. Their concessions are dependent on both the amount of available resources but also on their preferences.

Further work includes the development of resource-bounded strategies that may be useful for the m-commerce domain, the design of an agent's utility functions and dynamic levels of commitments associated to speech-acts (as mentioned in [24]). More negotiation simulations are being executed on our test-bed and future work includes more extensive analysis of the reliability of the algorithms and different strategies with respect ot the parameters and metrics in Section VI. Specifically, we aim to provide more performance results regarding various values for the quality of service of the network to show the agent's adaption using both the network-aware and the experience-aware strategies. We can also analyze more complex forms of evaluating and generating the responses in the various strategies through market-oriented mechanisms and refinement of the strategy $\mathcal{E}$ to analyze other forms of experience in m-service negotiations. Finally, we are also evaluating the robustness of m-service negotiation through synchronization layers for message exchange and belief revision in the agents [30].

ACKNOWLEDGMENT

This research has formed part of the Software Based Systems area of the Core 2 Research Programme of the Virtual Centre of Excellence in Mobile and Personal Communications, Mobile VCE. More detailed technical reports on this research are available to Industrial Members of Mobile VCE.

REFERENCES

[1] N. Sadeh, *M-Commerce: Technologies, Services, and Business Models*. New York: Wiley, 2002.

[2] N. R. Jennings, "An agent-based approach for building complex software systems," *Commun. ACM*, vol. 44, no. 4, pp. 35–41, 2001.

[3] M. Wooldridge and N. R. Jennings, "Intelligent agents: Theory and practice," *Knowl. Eng. Rev.*, vol. 10, no. 2, pp. 115–152, 1995.

[4] N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, C. Sierra, and M. Wooldridge, "Automated negotiation: Prospects, methods and challenges," *Int. J. Group Dec. Neg.*, vol. 10, no. 2, pp. 199–215, 2001.

[5] P. J. Turner *et al.*, "Scenarios for Future Communications Environments," ECS Dept., Southampton Univ., U.K., Tech. Rep. ECSTR-IAM02–005, 2002.

[6] S. Kraus, *Strategic Negotiation in Multiaagent Environments*. Cambridge, MA: MIT Press, 2001.

[7] J. Rosenschein and G. Zlotkin, *Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers*. Cambridge, MA: MIT Press, 1994.

[8] T. Sandholm and N. Vulkan, "Bargaining with deadlines," *AAAI*, pp. 44–51, 1999.

[9] Z. Maamar, "Agent technologies for m-services," presented at the Net. Object Days, Workshop Agent Technology for E-Services, 2002.

[10] i-mode. NTTDoCoMo. [Online]. Available: http://www.nttdocomo.com

[11] Y. Lashkari, M. Metral, and P. Maes, "Collaborative interface agents," presented at the *12th Nat. Conf. Artificial Intelligence*, Seattle, WA, 1994.

[12] Foundation for Intelligent Physical Agents, fipa. Agent Communication Language. FIPA. [Online]. Available: http://www.fipa.org

[13] J. R. Searle, *Speech Acts: An Essay in the Philosophy of Language*. Cambridge, U.K.: Cambridge Univ. Press, 1969.

[14] S. Paurobally and R. Cunningham, "Verification of protocols for negotiation between agents," in *ECAI-15*, 2002, pp. 43–48.

[15] D. Chalmers and M. Sloman, "A survey of quality of service in mobile computing environments," *IEEE Commun. Surv. Tutorials*, vol. 2, 1999.

[16] M. Wooldridge, *An Introduction to Multiagent Systems*. New York: Wiley, 2002.

[17] S. Paurobally and R. Cunningham, "Specifying the processes and states of negotiation," in *AMEC. The European AgentLink Perspective*. New York: Springer-Verlag, 2000, pp. 61–77.

[18] P. Faratin, C. Sierra, and N. R. Jennings, "Negotiation decision functions for autonomous agents," *Int. J. Robot. Auton. Syst.*, vol. 24, no. 3-4, pp. 159–182, 1998.

[19] D. Poole, A. Mackworth, and R. Goebel, *Computational Intelligence. A Logical Approach*. London, U.K.: Oxford Univ. Press, 1998.

[20] H. Raiffa, *The Art and Science of Negotiation*. Cambridge, MA: Harvard Univ. Press, 1982.

[21] P. Faratin, C. Sierra, and N. R. Jennings, "Using similarity criteria to make negotiation trade-offs," in *ICMAS*, 2000, pp. 119–126.

[22] S. Kraus, "Beliefs, time and incomplete information in multiple encounter negotiations among autonomous agents," *Ann. Math. AI*, vol. 20, no. 1-4, pp. 111–159, 1996.

[23] K. Larson and T. Sandholm, "An alternating offers bargaining model for computationally limited agents," presented at the AAMAS, Bolgna, Italy, 2002.

[24] S. Paurobally, P. J. Turner, and N. R. Jennings, "Toward automating negotiation for m-services," in *Proc. AMEC V Workshop, 2nd Int. Joint Conf. Autonomous Agents Multi-Agent Systems*, 2003, pp. 124–131.

[25] H. Chen, A. Joshi, and T. Finin, "Dynamic service discovery for mobile computing: Intelligent agents meet jini in the aether," *Cluster Computing, Kluwer*, vol. 4, no. 4, pp. 343–354, 2001.

[26] D. Chakraborty, F. Perich, S. Avancha, and A. Joshi, "Dreggie: Semantic service discovery for m-commerce applications," presented at the Semantic Service Discovery for M-Commerce Applications, Workshop Reliable Secure Applications Mobile Environment, 20th Symp. Reliable Distributed Systems, New Orleans, LA, 2001.

[27] T. Finin, A. Joshi, L. Kagal, O. Ratsimore, V. Korolev, and H. Chen, "Information agents for mobile and embedded devices," presented at the 5th Int. Workshop Cooperative Information Agents, Modena, Italy, 2001.

[28] S. Kraus, "Negotiation and cooperation in multi-agent environments," *Artif. Intell. J., Special Issue on Economic Principles of Multi-Agent Systems*, vol. 94, no. 1-2, pp. 79–98, 1997.

[29] S. Kraus, J. Wilkenfeld, and G. Zlotkin, "Multiagent negotiation under time constraints," *Artif. Intell.*, vol. 75, no. 2, pp. 297–345, 1995.

[30] S. Paurobally, R. Cunningham, and N. R. Jennings, "Ensuring consistency in joint beliefs of interacting agents," in *Proc. 2nd Int. Joint Conf. Autonomous Agents Multi-Agent Systems*, 2003, pp. 662–669.

Shamimabi Paurobally received the B.Eng. and Ph.D. degrees in computing from Imperial College, London, U.K., in 1996 and 2002, respectively.

She is currently a Research Fellow in the School of Electronics and Computer Science, Southampton University, U.K., and is working on a project, funded by the Mobile VCE consortium, to research automated negotiation in future generation mobile telecommunication environments. Her research interests include automated negotiation, m-commerce and facilitating agent interaction through formally specified and verified interaction protocols and through reasoning about the common beliefs of multiagent systems.

Phillip J. Turner received the B.Sc. degree in computing with real-time computer studies from the University of Gloucester, Gloucester, U.K. in 1993, the M.Sc. degree in intelligent knowledge-based systems from Essex University, U.K., in 1995. He is currently pursuing the Ph.D. degree in improving the scalability of multiagent systems through the use of self-organization, University of Southampton, Southampton, U.K.

He is currently a Research Fellow in the School of Electronics and Computer Science, Southampton University, U.K. This work was funded by the Mobile VCE consortium for which he researched large scale fault tolerant multiagent systems for future generation mobile telecommunications environments. He is currently working on a project funded by the DTI Next Wave Technologies and Markets program to research the trust issues that arise from accessing software services and utilizing agent technology in pervasive computing environments as part of a collaborative effort involving the QinetiQ Distributed Technology Group, together with the Declarative Systems and Software Engineering (DSSE) group, and the Intelligence Agents Multimedia (IAM) group, both at the University of Southampton. His research interests include scalability, self-building and self-configuring multiagent systems and agents, multiagent fault tolerance, risk management and reputation mechanisms, and distributed information management.

Nicholas R. Jennings is Professor of computer science in the School of Electronics and Computer Science, Southampton University, U.K., where he carries out basic and applied research in agent-based computing. He is Deputy Head of School (Research), Head of the Intelligence, Agents, Multimedia Group and is also the Chief Scientific Officer. He helped pioneer the use of agent-based techniques for real-world applications and he has also made theoretical contributions in the areas of automated negotiation and auctions, cooperative problem solving, and agent-oriented software engineering. He has published some 200 articles on various facets of agent-based computing, and is in the top 125 most cited computer scientists (according to the citeseer digital library).

Prof. Jennings has won several awards for his research including the Computers and Thought Award (1999), an IEE Achievement Medal (2000), and the ACM Autonomous Agents Research Award (2003).