E-Commerce Services based on Mobile Agents

Giancarlo Fortino, Alfredo Garro, Wilma Russo
DEIS-Università della Calabria, Italy
{g.fortino, garro, w.russo}@unical.it

INTRODUCTION

The Internet offers a unique opportunity for e-Commerce to take central stage in the rapidly growing online economy. With the advent of the Web, the first generation of Business-to-Consumer (B2C) applications was developed and deployed. Classical examples include virtual shops, on-demand delivery of contents, and e-travel agency. Another facet of e-Commerce is represented by Business-to-Business (B2B) which can have even more dramatic economic implications since it far exceeds B2C in both the volume of transactions and rate of growth. Examples of B2B applications include procurement, Customer Relationship Management (CRM), billing, accounting, human resources, supply chain, and manufacturing (Medjahed, Benatallah, Bouguettaya, Ngu & Elmagarmid, 2003).

Although the currently available web-based and object-oriented technologies are well-suited for developing and supporting e-Commerce services, new infrastructures are needed to achieve a higher degree of intelligence and automation of e-Commerce services. Such a new generation of e-Commerce services can be effectively developed and provided by combining the emerging Agent paradigm and technology with new web-based standards such as ebXML (ebXML, 2005).

Agents have already been demonstrated to retain the potential for fully supporting the development lifecycle of large-scale software systems which require complex interactions between autonomous distributed components (Luck, McBurney & Preist, 2004). In particular, e-Commerce has been one of the traditional arenas for agent technology (Sierra & Dignum, 2001). Agent-mediated e-Commerce (AMEC) is concerned with providing agent-based solutions which support different stages of the trading processes in e-Commerce including needs identification, product brokering, merchant brokering, contract negotiation and agreement, payment and delivery, and service and evaluation. In addition, the mobility characteristic of peculiar agents (aka mobile agents), which allows them to move across the nodes of a networked environment, can further extend the support offered by the Agents by featuring advanced e-Commerce solutions such as location aware shopping, mobile and networked comparison shopping, mobile auction bidding, and mobile contract negotiation (Kowalczyk, Ulieru & Unland, 2003; Maes, Guttman & Moukas, 1999).

To date, several agent- and mobile agent-based e-Commerce applications and systems have been developed which allow for the creation of complex e-
Marketplaces, i.e. e-Commerce environments which offer buyers and sellers new channels and business models for trading goods and services over the Internet.

However, the growing complexity of agent-based Marketplaces demands for proper methodologies and tools supporting the validation, evaluation and the comparison of: (i) models, mechanisms, policies and protocols of the agents involved in such e-Marketplaces; (ii) aspects concerning with the overall complex dynamics of the e-Marketplaces.

The use of such methodologies and tools can actually provide the twofold advantage of:

(i) analyzing existing e-Marketplaces to identify the best reusable solutions and/or identify hidden pitfalls for reverse engineering purposes;

(ii) analyzing new models of e-Marketplaces before their actual implementation and deployment to identify, a priori, the best solutions so saving reverse engineering efforts.

This article presents an overview of an approach to the modeling and analysis of agent-based e-Marketplaces (Fortino, Garro & Russo, 2005; Fortino, Garro & Russo, 2004). The approach centers on a Statecharts-based development process for agent-based applications and systems (Fortino, Russo & Zimeo, 2004) and on a discrete event simulation framework for mobile and multi agent systems (Fortino, Garro & Russo, 2004). A case study modeling and analyzing a real consumer-driven e-Commerce service system based on mobile agents within an agent-based e-Marketplace on the Internet (Bredin, Kotz & Rus, 1998; Wang, Tan & Ren, 2002) is also described to demonstrate the effectiveness of the proposed approach.

BACKGROUND

In a broad sense, an agent is any program that acts on behalf of a (human) user (Karnik & Triphati, 1998). An agent can just sit there and interact with its environment and with other agents through conventional means, such as local/remote procedure calls and asynchronous messaging, or through more advanced coordination infrastructures such as Tuple spaces and Event-based systems. Agents that do not or cannot move are called “stationary agents”. Conversely, a mobile agent is a program that represents a user in a computer network and can migrate autonomously from node to node to perform some computation on behalf of the user. Thus mobility is an orthogonal property of agents, that is, not all agents are mobile. Also mobile agents can interact with their environment and, notably, with other agents through mobility-aware and mobility-unaware infrastructures (Fortino & Russo, 2005). Indeed, the emergence of mobile agents was motivated by the benefits they provide for creating distributed systems. In fact, as Lange and Oshima pointed out in their seminal paper (Lange & Oshima, 1999), there are at least seven good reasons to employ mobile agents: reduction of network load, overcoming of network latency, encapsulation of protocols, asynchronous and autonomous execution (“dispatch your agents, shut off your machine”), dynamic adaptation, seamless system integration, and robustness and fault-tolerance.

An Agent-based e-Marketplace (AEM) is a distributed multi-agent system formed by both stationary and mobile agents which provide e-Commerce services to end-users within a business context. AEMs are, as previously pointed out, distributed large-scale complex systems which require tools which are able to
analyze not only the AEM at the micro level, i.e. behaviors and interactions of their constituting agents, but also the AEM at the macro level, i.e. the overall AEM dynamics.

In (Griss & Letsinger, 2000), an agent-based framework for e-Commerce simulation games has been developed by using Zeus, a Java-based multi-agent system developed at the British Telecom Lab. Its goal is to evaluate the potential consequences of novel combinations of market models, business strategies and new e-services through multi-player shopping games, in which agents represent various typologies of sellers, buyers, brokers and services.

In (Wang, Tan & Ren, 2002), an infrastructure for Internet e-Marketplaces, on the basis of Aglets mobile agents which enables real commercial activities by consumers, agents and merchants, has been proposed. Its goal is not only to provide an advanced e-Commerce service but also to evaluate several dispatching models for mobile agents.

In (Bredin, Kotz & Rus, 1998) a simulated environment for mobile agents is described which allows analyzing the market-based resource control system of the D’Agents mobile agent system and, in particular, the resource allocation mechanism of its resource manager using a sealed-bid second-price auction policy.

Although useful insights about AEM micro and macro levels can be acquired by playing e-Commerce simulation games and, then, analyzing the obtained results, or by evaluating real e-Commerce applications, discrete event simulators are essential for evaluating how AEMs work on scales much larger than that achievable in games or in applications which involve humans. In fact, discrete event simulation is currently extensively exploited as a strategic tool in most research and application areas which are directly or indirectly related to computer science. In this context, the article proposes an approach based on discrete event simulation and shows its application to the analysis of micro level issues of a consumer-driven AEM: validation and evaluation of services based on mobile agents for product searching and buying.

MODELING AND ANALYSIS OF MOBILE AGENT-BASED SYSTEMS

The Statecharts-based approach for modeling and analysis

The proposed approach (Fortino, Garro & Russo, 2005) consists of the following phases: High-Level Modeling, Detailed Design, Coding and Simulation (see Figure 1).
The **High-Level Modeling** of an agent-based system can be supported through well-established agent-oriented methodologies (such as the Gaia methodology; Wooldridge, Jennings & Kinny, 2000) which cover the phases of requirements capture, analysis and high-level design. An agent-based system (AS) can be modeled as follows:

\[
\text{AS} = \langle \text{AT, LCL, act, ser, pro} \rangle,
\]

where:

- **AT** (Agent Types) is the set of types of agents embodying activity, offering services and interacting with each other;
- **LCL** (Logical Communication Links) is the set of logical communication channels among agent types which embody interaction protocols;
- **act**: \( \text{AT} \rightarrow \text{activity description} \) is the activity relation which associates one or more activities to an agent type;
- **ser**: \( \text{AT} \rightarrow \text{service description} \) is the service relation which associates one or more services to an agent type;
- **pro**: \( \text{LCL} \rightarrow \text{interaction description} \) is the protocol relation which associates an interaction protocol to a logical communication channel.

The **Detailed Design** of an AS is achieved through a Statecharts-based formalism, namely Distilled StateCharts (DSC) (Fortino, Russo & Zimeo, 2004), which allows for the specification of the behavior of the agent types and the interaction protocols among the agent types. In fact, a Statecharts-based specification of an entity describes both internal behavior and coordination through the reception and generation of events (Harel & Gery, 1997). DSC allow for the specification of the behavior of lightweight agents which have the following features: event-driven, single-threaded, capable of transparent migration, and executing chains of atomic actions. The DSC-based specification of an AS (\( \text{AS}_{\text{DSC}} \)) can be expressed as follows:

\[
\text{AS}_{\text{DSC}} = \{ \text{Beh(AT)}_{1}, \ldots, \text{Beh(AT)}_{n} \},
\]

where:

- \( \text{Beh(AT)}_{i} \) is the DSC-based specification of the behavior of the \( i \)-th agent type. 
- \( \text{Beh(AT)}_{i} = \langle \text{S}_{\text{Beh(AT)}_{i}}, \text{E}_{\text{Beh(AT)}_{i}} \rangle \), where \( \text{S}_{\text{Beh(AT)}_{i}} \) is a hierarchical state machine incorporating the activity and interaction handling of the \( i \)-th agent type and \( \text{E}_{\text{Beh(AT)}_{i}} \) is the related set of events to be handled triggering state transitions in \( \text{S}_{\text{Beh(AT)}_{i}} \).

The **Coding** of an \( \text{AS}_{\text{DSC}} \), \( \text{C(AS}_{\text{DSC}}) \), is carried out through the Java-based Mobile Active Object Framework (MAO Framework; Fortino, Russo & Zimeo, 2004). In particular, \( \text{Beh(AT)}_{i} \) can be seamlessly translated into a composite object.
(called MAOBehavior object), which is the object-based representation of $S_{Beh}(AT_i)$, and into a set of related event objects representing $E_{Beh}(AT_i)$.

Finally, the Simulation phase of $ASP_{DSC}$ is supported by a Java-based discrete event simulation framework for distributed agent systems. The framework provides:

- **basic simulation objects**:
  - **Agent** ($Ag$), which represents a stationary or a mobile agent and includes a pair of objects: $<MAOId, MAOBehavior>$, where $MAOId$ is the unique agent identifier and $MAOBehavior$ is an agent behavior object;
  - **Events** ($Evt$), which represents the event for intra- and inter-$Ag$ interactions;
  - **AgentServer** ($AgS$), which represents the agent server hosting $Ag$s;
  - **VirtualNetwork** ($VN$), which represents the logical network of hosts on which $AgS$ are mapped;
  - **UserAgent** ($UA$), which represents a user, directly connected to an $AgS$, who can create, launch and interact with $Ag$s.

- a simulation engine enabling:
  - execution of $Ag$s by interleaving their $Evt$s processing;
  - transmission of $Evt$s among $Ag$s;
  - migration of $Ag$s;

On the basis of the framework, a simulator program can be implemented and executed to obtain a ResultSet containing validation traces and performance parameter values. While the validation of agent behaviors and interactions is carried out on execution traces automatically generated, the performance evaluation relies on the specific agent-based system to be analyzed; the performance evaluation parameters are therefore set ad-hoc. The ResultSet can also be used to feedback the high-level modeling and detailed design phases.

**A consumer-driven agent-based e-Marketplace**

A consumer-driven e-Marketplace is an e-Marketplace in which the exchange of goods is driven by the consumers that wish to buy a product. The modeled AEM, inspired by the systems presented in Bredin, Kotz, & Rus, 1998, and Wang, Tan, & Ren, 2002, consists of a set of stationary and mobile agents (see Figure 2) which provides basic services for the searching, buying, selling, and payment of goods.

The identified types of agents are:

- **User Assistant Agent** (UAA), which is associated with users and assists them in: (i) looking for a specific product that meets their needs; (ii) buying the product according to a specific buying policy.
- **Access Point Agent** (APA), which represents the entry point of the e-Marketplace. It accepts requests for buying a product from a registered UUA.
- **Mobile Consumer Agent** (MCA), which is an autonomous mobile agent dealing with the searching, contracting, evaluation, and payment of goods.
- **Vendor Agent** (VA), which represents the vendor of specific goods.
- **Yellow Pages Agent** (YPA), which represents the contact point of the distributed Yellow Pages Service (YPS) providing the location of agents selling a given product. The organization of the YPS can be: (i) **Centralized** (C), each YPA stores a complete list of Vendor Agents; (ii) **One Neighbor**
*Federated* (1NF), each YPA stores a list of VAs and keeps a reference to only another YPA; (iii) *M-Neighbors Federated* (MNF), each YPA stores a list of VAs and keeps a list of at most M YPAs.

- *Bank Agent* (BA), which represents a reference bank supervising money transactions between MCAs and VAs.

The identified types of interactions between the agent types are described below by relating them to the system workflow triggered by a user’s request (see Figure 2):

1. **Request Input** (UAA→APA): the UAA sends a request to the APA containing a set of parameters selected by the user for searching and buying the desired product, i.e. the product description (*Prod_Desc*), the maximum product price (*PMAX*) the user is willing to pay, and the type of buying policy (*BP*).
2. **Service Instantiation** (APA→MCA): the APA creates a specific MCA and provides it with the set of user’s parameters, the type of searching policy (*SP*), and the location of the initial YPA to be contacted. Upon creation, the MCA moves to the initial YPA location.
3. **Searching** (MCA↔YPA): the MCA requests a list of locations of VAs selling the desired product to the YPA. The YPA replies with a list of VA locations and, possibly, with a list of linked YPA locations.
4. **Contracting & Evaluation** (MCA↔VA): the MCA interacts with the found VAs to request an offer (*P_{offer}* ) for the desired product, evaluates the received offers, and selects an offer, if any, for which the price is acceptable (i.e., *P_{offer}*$\leq$*P_{MAX}*) according to the type of *BP*.
5. **Buying** (MCA↔VA↔BA): the MCA moves to the location of the selected VA and pays for the desired product using a given amount of e-cash (or bills) triggering the following money transaction: (i) the MCA gives the bills to the VA; (ii) the VA sends the bills to a BA; (iii) the BA validates the authenticity of the bills, disables them for re-use, and, finally, issues an amount of bills equal to that previously received to the VA; (iv) the VA notifies the MCA.
6. **Result Report** (MCA→UAA): the MCA reports the buying result to the UUA.

![Figure 2. The reference consumer-driven Agent-based e-Marketplace model: the types of agents, the logical communication links among them, and the sequence of agent interactions.](image)

**Analysis of mobile agent-based services**
A model of MCA is defined on the basis of the tuple: $\langle SP, BP, TEM \rangle$, where:

- **SP** is a searching policy in $\{\text{ALL, PA, OS}\}$:
  a. ALL: all YPAs are contacted;
  b. Partial (PA): a subset of YPAs are contacted;
  c. One-Shot (OS): only one YPA is contacted.

- **BP** is a buying policy in $\{\text{MP, FS, FT, RT}\}$:
  a. Minimum Price (MP): the MCA first interacts with all the VAs to look for the best price of the desired product; then, it buys the product from the VA offering the best acceptable price;
  b. First Shot (FS): the MCA interacts with the VAs until it obtains an offer for the product at an acceptable price; then, it buys the product;
  c. Fixed Trials (FT): the MCA interacts with a given number of VAs and buys the product from the VA which offers the best acceptable price;
  d. Random Trials (RT): the MCA interacts with a random number of VAs and buys the product from the VA which offers the best acceptable price.

- **TEM** is a task execution model in $\{\text{ITIN, PAR}\}$:
  a. Itinerary (ITIN): the Searching and Contracting & Evaluation phases are performed by a single MCA which fulfils its task by sequentially moving from one location to another;
  b. Parallel (PAR): the Searching and Contracting & Evaluation phases are performed by a set of mobile agents in a parallel mode. In particular, the MCA is able to generate a set of children (generically called workers) and to dispatch them to different locations; the workers can, in turn, spawn other workers.

Thus, each one of the defined models implements the product buying service differently. Figure 3 shows the generic DSC-based behavior of the MCA models $\langle \ast, \ast, \text{ITIN} \rangle$. For the sake of brevity, the explanation is not given here but readers unfamiliar with DSC-based programming can refer to (Fortino, Russo & Zimeo, 2004).
In order to analyze and compare the MCA models, the Task Completion Time \( (T_{TC}) \) parameter was defined as follows: \( T_{TC} = T_{CREATION} - T_{REPORT} \) where, \( T_{CREATION} \) is the creation time of the MCA and \( T_{REPORT} \) is the reception time of the MCA report. Accordingly, a simulator program was implemented which allows for computation of \( T_{TC} \) for each MCA model by varying the Yellow Pages organization, the number of YPAs \( (N_{YP}) \) and the number of VAs \( (N_{VA}) \). In particular, the simulation scenario was set up as follows:

- each stationary agent (UAA, APA, YPA, VA, BA) executes in a different agent server;
- agent servers are mapped onto different network nodes which are completely connected through links having the same characteristics. The communication delay \( (\delta) \) on a network link is modeled as a lognormally distributed random variable with a mean, \( \mu \), and, a standard deviation, \( \sigma \) (Floyd & Paxson, 2001);
- each UAA is connected to only one APA;
- the price of a product, which is uniformly distributed between a minimum \( (P_{MIN}) \) and a maximum \( (P_{MAX}) \) price, is set in each VA at initialization time and is never changed; thus the VAs adopt a fixed-pricing policy to sell products;
- each YPA manages a list of locations of VAs selling available products.
- an UAA searches for a desired product, which always exists in the e-Marketplace, and is willing to pay a price \( P_{MAX} \) for the desired product which can be any value uniformly distributed between \( P_{MAX} \) and \( (P_{MAX}+P_{MIN})/2 \).

Simulations were run by varying (i) the organization of the Yellow Pages (C, 1NF and 2NF organized as a binary tree or 2NFBT), (ii) the number of YPA agents in the range \([10..1000]\) and (iii) the number of VA agents in the range \([10..10000]\). These ranges were chosen for accommodating small as well as large e-Marketplaces. The duration of the performed simulations were set so to allow for the completion of the buying task carried out by the MCA.

Figure 4 shows the \( T_{TC} \) of the \(<,*,*\),ITIN> and \(<,*,*\),PAR> models in a medium-sized e-Marketplace with \( N_{YP}=10, N_{VA}=80, \) and YPS=2NFBT. The lowest-performance model is the \(<ALL,MP,ITIN>\) model. The \(<ALL,MP,*>\) models are the only models guaranteeing both a successful purchase and the best purchase since they are able to find the VA offering the minimum price. The \(<,*,*\),PAR> models always outperform the \(<,*,*\),ITIN> models but the \(<,*,FS,*>\) models where the \(<*,FS,ITIN>\) models perform similarly or slightly better than the \(<*,FS,PAR>\) models. However, in the latter case, purchase of the desired product at the best price is not guaranteed.
In order to compare the performances of PCA and ICA models, the results obtained for the <ALL, MP, * > MCA models adopting a YPA organization of the 2NFBT type are reported in Figure 5, where results were obtained setting $N_{YPA} = \{10, 100\}$ and varying $N_{VA}$. In agreement with the analytical model reported in (Wang, Tan & Ren, 2002), the PCA, due to its parallel dispatching mechanism, outperforms the ICA when $N_{VA}$ and $N_{YPA}$ are increased.

**FUTURE TRENDS**

To date **Agents** have been employed primarily for product and merchant discovery and brokering (Sierra & Dignum, 2001). The next stage will involve moving into real trading which will require considerable research and development efforts, including the definition,
implementation and, notably, analysis of new products and services such as market-specific shells, payment and contracting methods, risk assessment and coverage, quality and performance certification, security and trust management.

Moreover, in the very near future a rapid growth in agent-mediated auctions is expected. Auction is a long-established and well-understood trading mechanism, and the agent technology can be used to develop and support agent-mediated auction houses (Luck, McBurney & Preist, 2004).

In order to test these new trading and auction services within large-scale MAS, discrete-event simulation seems to be the most appropriate and reliable tool. Therefore, flexible and robust agent-oriented, discrete-event simulation frameworks must be carefully designed and developed to support analysis of a MAS at different levels of granularity: from agent behaviors, protocols and services (micro-level) to global MAS behavior (macro-level).

CONCLUSIONS

Using Agents to support e-Commerce (both B2C and B2B) is considered a key challenge for the agent community. This article has presented an integrated approach which effectively models and analyzes e-Commerce services based on Agents. In particular, a consumer-driven AEM was modeled and the product searching and buying strategies carried out by the mobile consumer agents in this AEM were analyzed. The consumer-driven AEM model used here was derived from real systems concerning with agent-based e-Marketplaces on the Internet (Bredin, Kotz & Rus, 1998; Wang, Tan & Ren, 2002). In line with the future trends which have been delineated, the proposed approach is being applied to the modeling and analysis of more complex AEMs and related services, and also enhanced by exploiting game theory and economics models.

REFERENCES


**Terms and Definitions**

**Agent:** an agent, in a broad sense, is any program that acts on behalf of a (human) user.

**Mobile Agent:** a mobile agent is a program that represents a user in a computer network and can migrate autonomously from node to node, to perform a computation on behalf of the user.

**ebXML:** *Electronic Business XML* is an XML-based language and infrastructure which aims at enabling B2B interactions among companies of any size.

**AMEC:** *Agent-Mediated e-Commerce* is concerned with providing agent-based solutions which support different stages of the trading processes in e-Commerce including needs identification, product brokering, merchant brokering, contract negotiation and agreement, payment and delivery, and service and evaluation.
**e-Marketplace**: an Electronic Marketplace is an e-Commerce environment which offers new channels and business models for buyers and sellers to trade goods and services over the Internet.

**AEM**: an Agent-based e-Marketplace is a distributed multi-agent system formed by stationary and mobile agents which provide e-Commerce services to end-users within a business context.

**Distilled StateCharts**: Distilled StateCharts are a Statecharts-based formalism for lightweight mobile agents.