MAST: an Agent Framework to Support B2C E-Commerce

Salvatore Garruzzo, Domenico Rosaci and Giuseppe M.L. Sarné
DIMET - University Mediterranea of Reggio Calabria
Località Feo di Vito - 89060 Reggio Calabria, Italy
Email: {salvatore.garruzzo, domenico.rosaci, sarne}@unirc.it

***Abstract*— In this paper we present an XML-based multi-agent system, called *Multi Agent System for Traders* (MAST), that completely supports Business-to-Customer E-commerce activities, including advertisements and payments. MAST helps both customers and merchants in their tasks with a homogeneous and personalized approach. In particular, E-payments in MAST are implemented under the availability of financial institutions. This avoids exchanging of sensible customers' information and reinforces the confidence between customers and merchants. A complete prototype of MAST has been implemented in the JADE framework, and it has been exploited for realizing some experiments, in order to evaluate its performances.**

I. Introduction

A great contribution to the Internet diffusion has been provided by E-Commerce (EC) activities, i.e. all trading activities carried out by means of Internet. In [15] a classification of the EC activities in homogeneous categories has been realized on the basis of the typology of the traders and of the specific trade activity carried out over the Internet. In this paper we deal with one of these categories, i.e. the Business-to-Consumer (B2C), that can be compared to the retail trade of traditional commerce.

Nowadays, the B2C involves a large number of merchants interested in offering products using a convenient media and customers that desire to purchase those products. In this context, customers and merchants can exploit different opportunities as: *(i)* absence of time and space boundaries; *(ii)* simple, fast and comfortable purchases; *(iii)* low costs and several sale terms available. However, a significant customer-merchant distrust still persists, mostly due to the absence of personal contacts and to a low acceptance of the e-payment methods for security reasons. To capture the different phases carried out by enacting a B2C process, some behavioural models can be exploited, and particularly, in this paper we adopt the Consumer Buying Behavior (CBB) model [8], where the trading activities have been embedded in six different phases (resp., "Need Identification", "Product Brokering", "Merchant Brokering", "Negotiation", "Purchase and Delivery", "Service and Evaluation").

This work presents a multi-agent framework to support B2C activities of merchants and customers. Such a framework, called *Multi-Agent System for Traders* (MAST), is composed of a set of agents and a central agency. In particular, in MAST each merchant and each customer is provided by a software agent, managing a personal profile, able to support B2C activities during all the CBB stages. The MAST framework presents the following important features: *(i)* software agents are XML-based to manage agent profiles and messages in a light and easy manner and to realize agent communications in ACML language [3], [7] assuring portability; *(ii)* an *Ontology* is used as a common language for all agents allows to unify the representation of products and categories belonging to various catalogues; *(iii)* an e-payment protocol called AIPP (Agent Internet Payment Protocol) [6], based on existing financial institutions, is fully compliant with the standard FAST [2] and it is used together with single-use account identifiers [18]; *(iv)* a central agency provides agents with some services and cooperates with them to realize only the "Need Identification" and the "Service and Evaluation" stages of the CBB model in an efficient way.

The paper is organized as follows: the MAST framework is presented in the following section. In Sect. III the AIPP protocol is briefly illustrated and in Sect. IV the adopted functionalities for customer and merchant support are described. In Sect. V the MAST prototype and performances are discussed. Section VI deals with some Related Work and finally, in Sect. VII, some conclusions are drawn.

II. The MAST Framework

In the MAST framework, represented in Fig. 1, each customer C and merchant M is associated to her/his personal agent (resp., c and m) and with her/his financial institution (FI). All agents are logged into the MAST Agency (Ag). Both agents and agency support B2C activities managing (in terms of insertion, deletion and updating) their respective *Knowledge* profiles. In this section, agents and agency will be briefly described by illustrating their profiles and behaviours, while the B2C support activities in the CBB stage are exposed in Sect. IV.

A. The MAST Agents

In the following, U denotes the generic user (a customer or a merchant) and a represents her/his agent. Each MAST agent manages its Agent Knowledge (AK) profile, represented in Fig. 2 and described by the following elements:

- UD (*User Data*), contains the user's name ($Name$) and address ($Address$), login identifier (AcL), password (AcP), real (Ac) and single-use (AcT) user's account

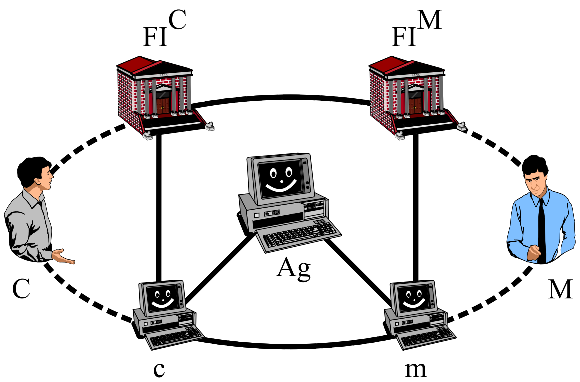

Fig. 1. The MAST architecture

identifiers, all referred to the user's account into FI. Note that Ac and AcT include also FI coordinates.

- AD (*Agent Data*), containing the user's agent (aI) and Agency (AgI) identifiers and a pruning threshold (T) to delete uninteresting information from AK.
- O (*Ontology*). In order to identify products of interest for the users, in our framework as Ontology we adopt the North America Industry Classification (NAICS) [13], which is an official hierarchical industrial classification used in North America, employing unified evaluation criteria. In MAST, the 6 digits NAICS code is used to identify a product. It is clear that other ontologies of this kind might as well be adopted instead of the NAICS coding.
- PD (*Product Data*) is a set of products, where each product is represented by an identifier (PI) and described by the following elements: NAICS code (N); model (M); brand (B); price (P); currency (C); commercial unit (U); auction flag (A) that is set to 1 if the product has a fixed price otherwise it is set to 0; tax (X); benefits set ($BSet$), eventually empty, of added values; delivery set ($DSet$), that collects the delivery identifiers (see DD section) related to the product.
- DD (*Delivery Data*) is a set of elements, where each element, represented by an identifier (DI), is described by the delivery time (DT) and by the fixed (F) and variable (V) costs. Note that DD collects the data of the chosen delivery for a customer, while it collects the data of the delivery he/she makes available for a merchant.
- ADB (*Agent Data Base*) is a set of agent (and Agency) data where each element, represented by its identifier aI, is described by its Internet address (aA) and by the date of its update (aAU).
- UP (*User Profile*) is a set of data that an agent a obtains monitoring the CBB activities in the MAST environment; for a customer its c agent collects the data of the products which the customer is interested in; elsewhere, for a merchant its m agent collects the data of the CBB activities carried out in the site by the agents of the various customers for the products offered by the merchant. Each element of UP is represented by the identifier PI (the same of the PD section) and it is described by the following elements: visit counter (VC); first visit (FV), one before the last visit (PV) and last visit (LV) dates; product rate (R); a set ($PASet$), where each element is associated to an agent that has been interested in the product, and that is composed by an agent identifier (aI), the highest CBB stage reached and eventually the delivery identifier (DI) and the auction flag (A). More in detail, the rate R represents the interest of a customer for a specific product and it is updated (by a) when a CBB activity is monitored by a using the following formula:
$$R = \phi \cdot \sum_{\xi=1}^{5}\left[\frac{CV\cdot(1+PV-FV)}{((6-\xi)^2\cdot(1+LV-PV))}\right]$$
where: $\xi \in [1, \cdots, 5]$ identifies one among the first five CBB stages; $\phi \in \{1; -1\}$ describes the satisfaction of the customer about a product (ϕ is usually set to 1, but it is set to -1 by the customer only if in the last and optional CBB stage he/she is unsatisfied of the purchased product; for a merchant, ϕ is always set to 1). Furthermore, the differences among FV, PV and LV are expressed in days[1] beginning from FV. Note that R depends on the number of times that an activity has occurred in a CBB stage, the relevance of the involved CBB stage and the more recent accesses.

The information in the described structures are used by an agent a to realize its goals, as explained in the following, excluding the CBB support which is presented in Sect. IV. More in detail:

- ***setup steps***: semi-automatic procedures are activated to: *(i)* set initially or update UD, AD, ADB, interacting with Ag when it is needed as in the first a's activation; *(ii)* remove a from the system for an U's request to Ag.
- ***operational steps***: a customer agent is automatically activated (resp., deactivated) when a Web session starts (resp., ends "per se" or for an explicit customer's choice), whereas a merchant agent is automatically activated (resp., deactivated) when its site is on-line (resp., off-line or for an explicit merchant's choice). An agent performs the following activities: *(i)* it sends periodically to Ag its aA; *(ii)* it constructs its profile to support its user updating its AK w.r.t. each agent contact and each access for a product in one or more CBB stages; *(iii)* in order to realize the first phase of the "Need Identification" CBB stage, in MAST a customer agent periodically sends to Ag a list (L) containing the NAICS code of those products that meet interests and preferences of its user, ordered on the basis of their rate R; *(iv)* periodically each agent prunes its AK from some evaluated unimportant information on the basis of the values of the rating w.r.t. the threshold T.

B. The MAST Agency

The Agency Knowledge (AgK) profile is described by:

- AgD (*Agency Data*), that is composed by the Agency Identifier (AgI) and Internet Address (AgA).

[1]The choice of the day as reference time unit is due to the characteristics of the problem, given that purchases usually do not occur often in time (e.g., each minute or hour). However, it is possible to change the reference time unit without influencing the generality of the model.

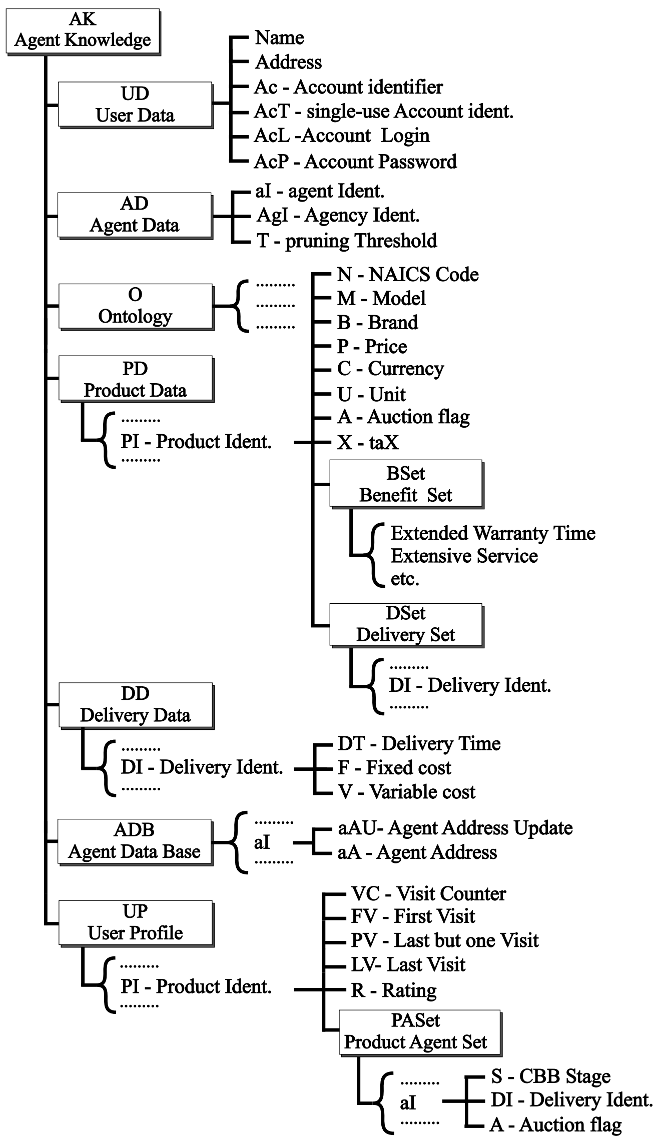

Fig. 2. The Agent Knowledge (AK)

- ADB (*Agent Data Base*), that is a set of agent data where each element, represented by its identifier aI, is described by: an Internet address (aA); the date of the aA update (aAU); a list (L) of NAICS code referred to those products of interest or preferences; the name ($Name$) and address ($Address$) of the agent's owner.
- aPT (*Agent Pruning Threshold*) that is exploited to deallocate long-time inactive agents.

The behaviour of the Agency consists of:

- ***affiliate managing steps***: The Agency carries out the following operations automatically: *(i)* when it is required, the Agency affiliates an agent sending it its *Identifier* and at this point the agent is logged and operative; *(ii)* the Agency updates the agent data when it changes; *(iii)* the Agency stores, for each active agent, the current address and the list L that the agent periodically sends to Ag; *(iv)* if a user requires an agent deletion to the Agency or if an agent is inactive for a time longer than the pruning threshold aPT, then the Agency deletes the agent and informs the community.
- ***service managing steps***: The Agency provides some services to agents, namely: *(i)* the support to realize some CBB stages efficiently as will be described in the following; *(ii)* a broadcasting message service (e.g., to provide an agent m offer to all agents c); *(iii)* a yellow page service, where each affiliate can ask the address of another MAST affiliate.

III. The Agent Internet Payment Protocol (AIPP)

Payment schemes can be assessable with subjective criteria, as customer acceptance or trust [14], and objective criteria, as functionality and quality parameters like transaction cost, security, privacy, etc. The presence of a network in a payment scheme introduces new issues, absent in traditional scenarios [1], [16], where: *(i)* identities of the transaction actors need to be authenticated and validated; *(ii)* payments and their effects guaranteed; *(iii)* operations, frauds and legal risks minimized. In addition, an extended use of standard protocols, existing products and services, payer anonymity, purchases confidentiality and low costs are desirable.

Currently, the most used e-payment system is the credit card, but in this case a credit card number should be provided to the merchant; this could be risky because the card number is provided over Internet and/or stored in the merchant site. The electronic cash systems cannot be used due to law and crime prevention regulation/legislation [1]. Recently, centralized account schemes have grew quickly in popularity for their aptitude to integrate usual financial instruments in a secure Internet transaction context. This payment family, also proposed by well known financial institutions, includes general purpose or e-commerce specific applications and can be realized completely either in secure software or in secure hardware.

A centralized account approach has been proposed in 1999 by the Financial Service Technology Consortium (FSTC) with the Financial Agents Secure Transaction (FAST) project [5]. The FAST team has developed five payment schemes for different scenarios (without specifying any detailed protocol) based on financial institutions that manage user's accounts and agent technologies to take advantage from existing infrastructures. The main benefits are: *(i)* customers and merchants with no common authentication mechanisms (FAST is not an authentication model) are reciprocally authenticated by their financial institutions when they log in their on-line accounts with the usual procedures (commonly with login and password over an SSL connection [4]); *(ii)* payments occur directly via financial institutions to guarantee effective funds availability, funds transfer and connected effects, but also promote credit-push; *(iii)* interoperability among accounts located in different financial institutions is easy to effect (as between two banks) choosing among different transfer modalities usually available. On the contrary, it is hard when accounts are located into competitor payment systems; *(iv)* payments are carried out by agents that replace customers and merchants in most uninteresting and/or complex tasks.

The risks in FAST can be further minimized by transferring funds over interbanking networks, assigning to each message a time to live and a unique identifier, managing as much sensible information as possible off-line, etc. The problems of security communication among financial institutions, as those related to defense against viruses or hacker attacks, are beyond the FAST project objectives.

In this paper, we exploit the e-payment protocol AIPP (Agent Internet Payment Protocol) [6], complying with the FAST "pre-negotiation" scheme [2], together with single-use account identifiers [18]. In AIPP a low amount of information is exchanged without any explicit encryption level. Moreover, AIPP adopts only asynchronous agent communications without multiple Internet connections (other parties connected to the infrastructure, such as Internet providers, are considered as external risk factors). In this way, it is proposed as a potentially well acceptable Internet financial transaction method able to satisfy all issues of an e-payment scheme that have been previously described.

IV. The MAST Support to CBB Activities

MAST provides a support, in accordance to the CBB model, to customers and merchants in their EC activities. In MAST, typical interaction between agents involves a customer (C) with her/his agent (c) and financial institution (FI^C), a merchant (M) with her/his agent (m) and financial institution (FI^M), the Agency (Ag), a product (G) offered by a M. The appropriate financial institution typologies are limited to banks, card issuers or relevant financial organizations; further, it is assumed that payers and payees can manage their on-line accounts. In the following, the terms product and service will be used interchangeable.

In MAST, to avoid possible attacks, single-use account identifiers (preserving also financial privacy) and a nonce (i.e., an agent sender marker) are adopted, and a Time To Live (TTL) is used as message deadline for each agent communication. Moreover, to promote trust among customers and merchants, the AIPP protocol allows the FIs to be third parties in a financial transaction, still guaranteeing user's privacy.

Notation and data contents of the messages used in MAST to transfer in a consistent and efficient way the business information are illustrated before describing the MAST protocol. Note that the subscripts identify sender and receiver while *data* is an XML document[2], whose content is context sensitive (see Table I). More in detail:

- $INF_{x,y}(data)$: it requires/provides commercial information about a product;
- $REQ\_INV_{c,m}(data)$: it requires an invoice for a product offered by M;
- $INV_{m,c}(data)$: it contains the invoice required with $REQ\_INV_{c,m}(data)$;
- $PO_{c,m}(data)$: it is the purchase order w.r.t. $INV_{m,c}(data)$;

TABLE I
MESSAGE SPECIFICATION

Message	Message Content
$INF_{x,y}$	$\mathbf{H}(aS^x, aR^x, nc^x, TTL^x)$, $\mathbf{G}(N, M, B, P, C, U, A, X, BSet, DD, CUR, FP)$
$REQ\_INV_{c,m}$	$\mathbf{H}(aS^c, aR^c, nc^c, TTL^c)$, $\mathbf{G}(PI^M, N, M, B, P, C, U, A, X, BSet, DI^M, CUR, FP)$
$INV_{m,c}$	$\mathbf{H}(aS^m, aR^m, nc^m, TTL^m, PII^m), \mathbf{G}(PI^M, N, M, B, P, C, U, A, X, BSet, DI^M, DD, F, V, CUR, FP), \mathbf{F}(FII^M, FIA^M, AcT^M)$
$PO_{c,m}$	$\mathbf{H}(aS^c, aR^c, nc^c, TTL^c, PII^m)$, $\mathbf{F}(FII^C, FIA^C, AcT^C, Address^C)$
$PE_{x,y}$	$\mathbf{H}(aS^x, aR^x, nc^x, TTL^x, PII^m)$
$PA_{x,y}$	$\mathbf{H}(aS^x, aR^x, nc^x, TTL^x, PII^m)$
MTO_{c,FI^C}	$\mathbf{H}(aS^c, aR^c, nc^c, TTL^c, PII^m)$, $\mathbf{F}(FII^M, FIA^M, AcT^M, H(INV_{m,c}))$
$A\_MTO_{FI^C,c}$	$\mathbf{H}(aS^{FI^C}, aR^{FI^C}, nc^{FI^C}, TTL^{FI^C}, PII^m)$
$R\_MTO_{FI^C,c}$	$\mathbf{H}(aS^{FI^C}, aR^{FI^C}, nc^{FI^C}, TTL^{FI^C}, PII^m)$
$ACT\_COD_{x,y}$	$\mathbf{H}(aS^x, aR^x, nc^x, TTL^x, PII^m), \mathbf{F}(H(INV_{m,c}))$
$NEW\_AI_{x,y}$	$\mathbf{H}(aS^x, aR^x, nc^x, TTL^x), \mathbf{F}(AcT^y)$
$EVAL_{c,m}$	$\mathbf{H}(aS^c, aR^c, nc^c, TTL^c, PII^m)$

In the first three CBB stage the messages can be addressed to c agents chosen among those listed in ADB or employing the Ag's broadcasting messages service

- $PE_{x,y}(data)$ (resp., $PA_{x,y}(data)$): it notifies that the payment has been performed (resp., aborted) w.r.t. $PO_{c,m}(data)$;
- $MTO_{c,FI^C}(data)$: it is an irrevocable money transfer order w.r.t. $INV_{m,c}(data)$;
- $A\_MTO_{c,FI^C}(data)$ (resp., $R\_MTO_{c,FI^C}(data)$): it notifies the MTO acceptance (resp., rejection) w.r.t. $MTO_{c,FI^C}(data)$;
- $ACT\_COD_{x,y}(data)$: it contains the MTO activation code w.r.t. $INV_{m,c}(data)$;
- $NEW\_AI_{x,y}(data)$: it contains a new single-use account identifier to be employed in the next purchase or sell;
- $EVAL_{c,m}(data)$: it is an optional evaluation of a purchase.

A *data* XML document is structured in three sections including:

1) *H (Header)* that is composed by: agent identifiers of *Sender* (aS) and *Receiver* (aR); *CBB Stage* (S); *Nonce* (nc) that is an agent's marker; *Time To Live* (TTL); *Product Invoice Identifier* (PII).
2) *G (Products)* that encodes: *Product Identifier* (PI) and the product data ($N, M, B, P, C, U, A, X, BSet$) previously described in Sect. II-A; one or more *Delivery Identifier*s (DI) with the corresponding data (DD, F, V), previously described in Sect. II-A; *Commercial Unit Required* (CUR); *Final Price* (FP).
3) *F (Financial)* is constituted by: *Financial Institution Identifier* (FII); *Financial Institution Internet Address* (FIA); *Financial Institution Single-Use Account identifier* (Ac); *User Address* ($Address$).

The actions performed by agents in MAST to support customers and merchants in their B2C activities during all CBB stages are described below in detail for each CBB stage and represented in Fig. 3.

a) Need Identification Support: ($\xi = 1$). In the first CBB stage, customers identify their needs and merchants advertise their offered products (G) to as more potential customers as possible. In detail: *(i)* when an M wants to make an offer about a product to potential Cs, he/she has to submit own offer sends to Ag an $INF_{m,c}$; *(ii)* in a first phase the Ag, on

[2]MAST agents employ the eXtensible Markup Language (XML) [22] to overcome several heterogeneity problems (platforms, languages, applications and communication modalities) and to transfer business information in a consistent way. Note that specific agreements must be established on the tag semantic.

the basis of the lists L provided by the c agents, takes care of sending the offer to the potentially appropriate c agents, then in a second phase the c agents present such offers to their Cs only if they are fully compatible with their interests and preferences.

b) Product Brokering Support: ($\xi = 2$). This stage occurs when a customer has identified a need and looks for a suitable product to satisfy it. In detail: *(i)* C can ask information on the desired product typology to one or more Ms by means of $INF_{c,m}$; *(ii)* all Ms that have a product that matches the C request, reply with a new $INF_{m,c}$ with all the details of the products and commercial information.

c) Merchant Brokering Support: ($\xi = 3$). A customer identifies the most suitable merchant to purchase a product carrying out the following actions: *(i)* If C has sufficient knowledge of the product details, a c's message $INF_{c,m}$ is sent to one or more merchants; *(ii)* if there is a product that matches C's request, M replies with a message that reports a complete description of the product; in such a way c can select the best product offer. Note that if in the previous stage C has received a sufficient number of $INF_{m,c}$ it is possible to choose a merchant without carrying out this present stage explicitly.

d) Negotiation Support: ($\xi = 4$). In this stage a pair of customer and merchant define the purchase details. They realize suitable strategies in a multi-round session for their respective bids and offers presented by means of messages. This stage is closed when an agreement is reached or the timeout TTL of the last message has elapsed.

e) Purchase and Delivery Support: ($\xi = 5$). In this stage the customer purchases, pays and chooses a delivery modality for a product offered by a merchant employing the AIPP protocol where: payer and payee identities are authenticated by their respective financial institutions during their on-line accounts accesses (usually with login and password over a *SSL Internet session*); payments occur directly among the financial institutions; Single-use account identifiers are adopted; No heavy protocol is needed; no sensible financial and commercial information is exchanged to assure privacy; financial institutions are third parties in the transaction to guarantee customers and merchants. The actions performed in this stage are: *(i)* When C wants to purchase a product offered by M, he/she sends the message $REQ\_INV_{c,m}$; *(ii)* m replies with $INV_{m,c}$ (a pro-forma invoice); *(iii)* c logs into FI^C and then orders a *Money Transfer Order* (MTO_{c,FI^C}) to FI^M payee; *(iv)* FI^C accepts/rejects the MTO on the basis of the existence of sufficient C's funds and notifies to c its choice with a $A\_MTO_{FI^C,c}$ + a new single-use account identifier (AcT) for the next purchase or with a $R\_MTO_{FI^C,c}$ message; *(v)* c sends a $PO_{c,m}$ to effect the purchase order; *(vi)* m logs into FI^M and sends the required payment activation code ($H(INV_{m,c})$ to FI^M; *(vii)* FI^M provides M with a new single-use account identifier (AcT) for the next sell and sends to FI^C the payment code ($H(INV_{m,c})$; *(viii)* if the activation code is the same as that provided by c, then FI^C effects the payment via FI^M and informs c about the state of success ($PE_{FI^C,c}$) or failure ($PA_{FII^C,c}$) of the MTO process; *(ix)* if the payment has been performed by FI^C, then FI^M informs m with a $PE_{FI^M,m}$ message, otherwise after the TTL of the $ACT\_COD_{FI^M,FI^C}$ message FI^M informs m with a $PA_{FI^M,m}$ of the sell failure; *(x)* finally, m could however accept the payment informing FI^M, and consequently FI^C, or refuse it aborting the sale and returns back the money to FI^C by means of its FI^M. At last FI^C will inform c whether the product has been purchased or not.

f) Service and Evaluation: ($\xi = 6$). It is an optional feedback provided by a customer to express her/his dissatisfaction about the purchase of a product, the merchant or both. Two kinds of actions can be carried out by the customer: *(i)* if the purchased product has been evaluated negatively, the Rate $R \in AK$ will assume a negative value by setting the ϕ coefficient to -1; *(ii)* if the merchant has been evaluated negatively, its identifier will be deleted from $PASet$ (w.r.t. G),

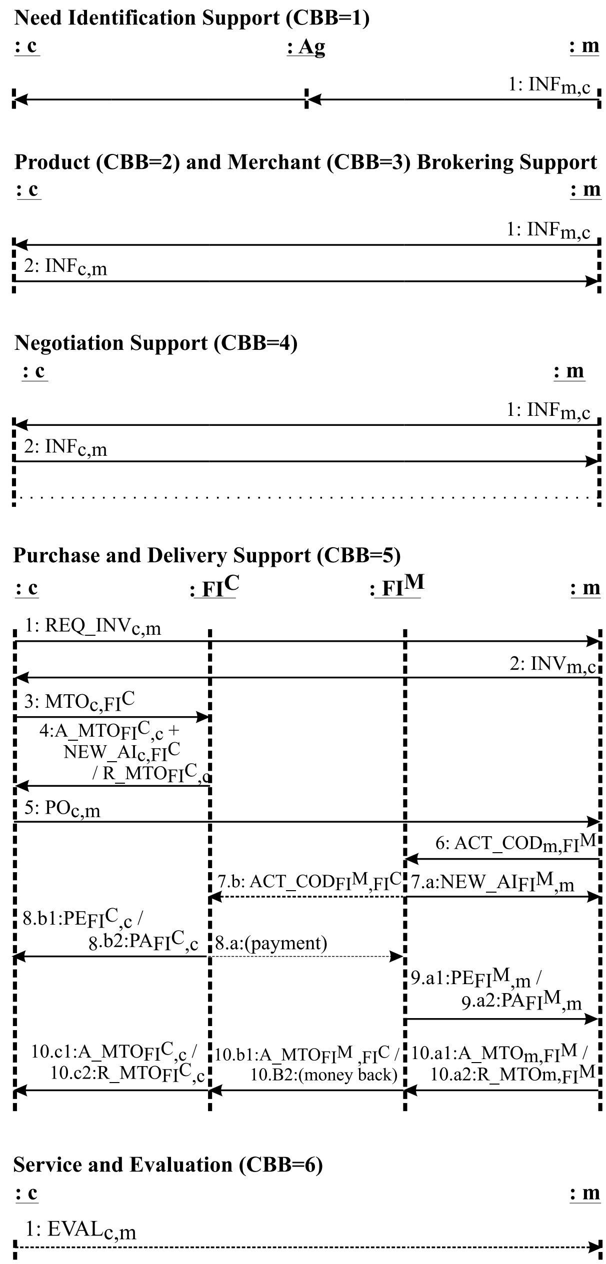

Fig. 3. UML of the MAST support activities

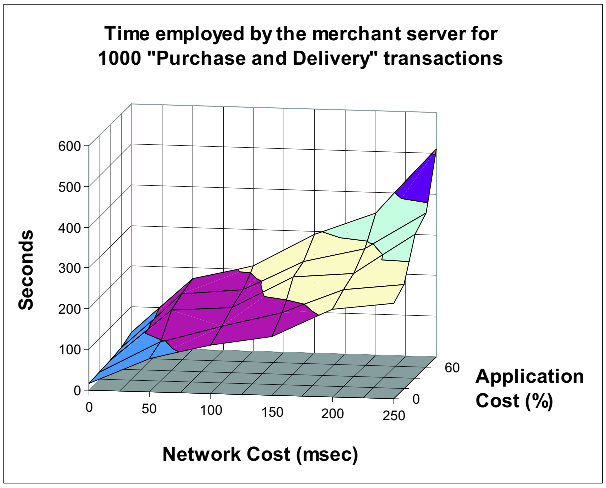

Fig. 4. The payment performances

then C might decide to inform M directly, or in an anonymous form by means of Ag, about her/his dissatisfaction.

V. System Prototype and Experiments

A complete prototype of MAST framework has been implemented in JADE [10], to test its CBB support activities simulating either single or continued sequences of CBB processes in a small B2C scenario[3]. Furthermore, to realize such experiments some EC sites have been realized in XML. In particular, in this section the results of the "Need Identification" and "Purchase and Delivery" stages are reported.

In the "Need Identification" the experiments have been finalized to measure the customer satisfaction degree (CSD) computed as the number of merchants' offers evaluated by the customers as correctly filtered by the system. The filtering process is carried out for each customer in two phases, the first performed by the Agency and finalized to disregard immediately all merchants' offers clearly out from the customer's interests and preferences; while the second phase is performed by the customer's agent to realize a fine tuning of the filtering activity on the basis of its profile. The tests have been carried out by 19 customers, using their agent profiles previously built on the basis of CBB activities carried out on XML EC sites.

More in detail, in the first phase on a total of 4750 merchant offers (250 offers for each customer), randomly chosen among the products offered in the XML EC sites, the agency has correctly rejected 3154 of them and considered potentially interesting 1596 offers. Then in the second phase, on the remaining 1596 offers the customers' agents have evaluated surely interesting only 193 offers, but 79 of them were not really interesting for the customers. In this way, we obtain a global CSD equal to 0,983. Note that we have set the filtering parameters to avoid the rejecting useful merchants offers.

About the "Purchase and Delivery" stage, it is clear that the cost located on the customers' client side has a minimal impact on the computational performance, since usually only one MAST activity runs at a time. Conversely, a merchant agent is associated with high computational cost, given that it has to satisfy a large volume of processes at the same time referring to different c agents. Consider that the server has to carry out other tasks for its EC activities that can absorb also a significant amount of resources and that have been simulated by assuming some different application costs as percentage of all processes carried out by an m agent; besides, the Internet cost can influence the global system performance and it has been simulated by setting some delays in the communications (tests were carried out on a 100 Mb LAN).

This procedure is surely a critical test activity, for the necessity to coordinate more parties, sending more messages and realizing some secure connections (SSL connections have been adopted in the tests). In the following, we employ this process only in order to obtain a rough estimation of the MAST computational efficiency in this CBB stage (keeping in mind that the MAST approach has anyway other significant benefits provided by the authentication mechanism and payment security level). In particular, the time necessary by the merchant's server to complete a sequence of 1000 "Purchase and Delivery" transactions for different application costs and network delays is represented in Fig. 4. Note that some steps occur using no secure connections and other occur on a simulated reserved banking channel.

The experiments suggest some considerations about the implemented MAST prototype and experimental results obtained. Using the MAST prototype, all CBB activities have been correctly carried out, customer and merchant profiles have been initialized, correctly updated and all our project goals have been meet, showing the capability of MAST to provide proper support to customers and merchants in EC scenarios. The experimental results obtained, even though they have only an indicative meaning both for the initial scenario assumption and some compulsory rough simulation show interesting performances in terms of efficiency, effectiveness and time employed.

VI. Related Work

The various aspects connected to the B2C have been dealt with in a very large variety of scientific works; some works which to our knowledge come closest to the material presented in this paper will be mentioned in this section.

The role of software agents in the EC has become very relevant, as proved by the large number of models and architectures proposed in literature and the state of the art has been investigated in a significant number of surveys [8], [9], [11], [12], [15], [19], [23]. The main opportunity offered by multi-agent systems is to support customers and merchants in performing their B2C activities. In the CBB context, MASs were traditionally focused on only in a few stages, usually "Merchant Brokering" and/or "Negotiation" stages, but progressively their support has been extended to all CBB stages (note that many MASs for B2C do not explicitly use the CBB model, but their activities are easily brought back to it). Furthermore, only a restricted number of MASs adopt one or more existent payment schemes explicitly, while the

[3]The simulations have been realized by employing computers based on single CPU (Intel Pentium 4, 3 GHz), RAM 2 Gbyte and O.S. Linux.

largest part of them just ignore the issue or record that a payment has occurred. Finally, since there is a large variety of protocols and communication languages that MASs adopt in B2C, these will not be specifically addressed here. In particular we propose the following three approaches:

- MAGMA [20] proposes a MAS for a free-market architecture based on messages. In MAGMA the agents are monitored by a central administration, only partially automatized; agents provide some trading strategies, independently by the users' behaviour, and can form agent alliances. Financial services for EC activities are provided, in a secure way, by a virtual bank that manages specific agents' account.
- CASBA [21], resulting from a CEE ESPRIT Project, implements an Internet agent-based marketplace supporting all CBB stages, in a flexible way with different auction types and dynamic negotiations, and some commercial payment schemas. CASBA employs Java, JavaScript, CORBA and XML technologies, while advertising is e-mail based. XML eases matching the data structures of the CASBA ontology with those of the client database.
- In [17] a model representing ontologies in a B2C scenario is proposed and a multi-agent architecture based on such model is described. It realizes a virtual marketplace agent-based where customers and merchants are supported by exploiting a representation of the concepts and behaviour involved in their EC world. Here, an agency has a central role as mediator in coalition formation, agent communications and in virtual auctions. No payment scheme is supported.

This aforementioned work exploit multi-agent systems to support B2C activities. They are explicitly CBB based or partially consistent with it. Customer interests and behaviour are taken into account in [17] where, similarly to CASBA, authors exploit XML to realize a unified representation in order to reduce the impact of heterogeneities. Payment issues are handled in MAGMA and in CASBA differently from MAST. Finally, MAGMA is designed to support also heterogeneous agents whereas CASBA, [17] and MAST deal only with homogeneous agents.

VII. Conclusion

This paper describes MAST, an XML-based multi-agent system to support customers and merchants in a suitable, homogeneous and personalized way, taking into account their interests on the basis of the behaviours shown during their B2C activities, represented as in CBB model. Furthermore, the opportunities offered by XML (for agent profiles, the messages, the inter-catalogue representation of products and categories and the agent communication language) are used along with those of a secure centralized payment scheme, based on existing financial institutions and single-use account numbers (payments happen only among financial institutions, over reserved communication channels, by preserving financial anonymity and confidentiality and benefiting of an existing authentication mechanism). Some results of experimental simulations in a small B2C scenario, carried out using a Jade-based prototypal implementation of MAST, are presented.

As for ongoing research, a development of MAST is planned by the introduction of different behavioural models taking in account emerging behaviours in the B2C area, such as formation of coalitions or the EC-site visiting.

References

[1] Asokan N., Janson P., Steiner M. and Waidner M. The State of the Art in Electronic Payment Systems. *IEEE Computer Magazine*, 30(9):28–35, September 1997.

[2] Financial Services Technology Consortium. Financial Agent Secure Transaction (FAST), Phase 1 Final Report (White Paper). FAST_Phase_1_White_Paper.pdf, September 2000.

[3] Foundation for Intelligent Physical Agents (FIPA). FIPA ACL Message Structure Specification. SC00061G.pdf, December 2005.

[4] Freier A.O., Karlton P. and Kocher P.C. The SSL Protocol version 3. ssl-toc.html, November 1996.

[5] The Financial Services Technology Consortium (FSTC) website, 2006.

[6] Garruzzo S., Sarné G.M.L. and Palopoli L. AIPP: A FAST-Complied E-Payment Protocol. In *IADIS International Conference - Applied Computing 2006 (IADIS'06)*, pages 406–410, San Sebastian, Spain, April 2006. IADIS Press.

[7] Grosof B. and Labrou Y. An Approach to using XML and a Rule-based Content Language with an Agent Communication Language. In *Issues in Agent Communication*, number 1916 in Lecture Notes in Computer Science, pages 96–117. Springer-Verlag, Heildeber, Germany, 2000.

[8] Guttman R.H., Moukas A.G. and Maes P. Agent-Mediated Electronic Commerce: A Survey. *The Knowledge Engineering Review*, 13(2):147–159, 1998.

[9] He M., Jennings N. R. and Leung H. On Agent-mediated Electronic Commerce. *IEEE Transaction on Knowledge and Data Engineering*, 15(4):985–1003, 2003.

[10] The Java Agent DEvelopment (JADE) framework website, 2006.

[11] Jennings N.R. and Wooldrige M.J. Applying Agent Technology. *Int. Journal of Applied Artificial Intelligence*, 9(4):351–361, 1995.

[12] Liu J. and Ye Y. Introduction to E-Commerce Agents: Marketplace Solutions, Security Issues, and Supply and Demand. In *Proc. of the E-Commerce Agents: Marketplace Solutions, Security Issues and Supply and Demand*, volume 2033 of *Lecture Notes in Computer Science*, pages 1–6. Springer-Verlag, London, UK, 2003.

[13] The North America Industry Classifications (NAICS) website, 2006.

[14] O'Mahony D., Pierce M. and Tewari H. *Electronic Payment Systems for E-Commerce*. Artech House, Norwood, MA, 2nd edition, 2001.

[15] Palopoli L., Rosaci D. and Ursino D. Agent's roles in B2C E-commerce. *AI Communications*, 1912:95–126, 2006.

[16] Pinheiro R. Preventing Identity Theft Using Trusted Authenticatorss. *Journal of Economic Crime Management*, 2(1):1–16, 2004.

[17] Rosaci D. A model of agent ontologies for B2C E-Commerce. In *Proc. of the Sixth International Conference on Enterprise Information Systems (ICEIS'04)*, pages 3–9, Porto, Portugal, 2004.

[18] Shamir A. SecureClick: A Web Payment System with Disposable Credit Card Numbers. In *Proc. of the 5th International Conference on Financial Cryptography*, volume 2339 of *Lecture Notes in Computer Science*, pages 232–242. Springer-Verlag, London, UK, 2002.

[19] Sierra C. and Dignum F. Agent-mediated Electronic Commerce. Scientific and Technological Roadmap. In *Agent Mediated Electronic Commerce. The European AgentLink Perspective*, volume 1991 of *Lecture Notes in Artificial Intelligence*, pages 1–18. Springer, 2001.

[20] Tsvetovatyy M.B. and Gini M. Toward a Virtual Marketplace: Architectures and Strategies. In *Proc. of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'96)*, pages 597–613, London, United Kingdom, April 1996.

[21] Vetter M. and Pitsch S. Towards a Flexible Trading Process over the Internet. In *Agent Mediated Electronic Commerce, The European AgentLink Perspective*, volume 1991 of *Lecture Notes in Computer Science*, pages 148–162. Springer-Verlag, London, UK, 2001.

[22] W3C. Extensible Markup Language (XML) 1.1, February 2004.

[23] Ye Y., Liu J. and Moukas A.G. Agents in Electronic Commerce - Special Issue on Agents in Electronic Commerce. *Electronic Commerce Research*, 1(1-2):9–14, 2001.