Symmetric Adaptive Customer Modeling for Electronic Commerce in a Distributed Environment

Maria Barra, Giuseppe Cattaneo, Mario Izzo, Alberto Negro, and Vittorio Scarano

Dipartimento di Informatica ed Applicazioni "R.M. Capocelli"
Università di Salerno, 84081 Baronissi (SA) – Italy

Abstract. Electronic Commerce involves more than simple online transactions, it also deals with several other fields such as market research, customers care, document exchange and customer modeling.
Our paper deals with the customer modeling issue, first introduced in the companion paper [4], where a model is presented of a system that can adapt its response to customers.
We present here the necessary extension to the model so that it can be used on a distributed system for EC on the World Wide Web.
The extensions presented are simple, easy to implement, not computationally expensive, and preserve the privacy of all the actors in an EC distributed system, while ensuring the adaptive behaviour.

1 Introduction

Electronic Commerce (EC) is getting much interest as research field, due to the development and widespread usage of the Internet and the World Wide Web. The sky-rocketing increase in number of users connected to the network makes this field central and important both from the scientifical and the commercial point of view. Many enterprises are offering services and goods through the network to an always wider audience and the challenge to the researchers is to develop and study distributed systems that are able to fulfill the promises of such "revolution".

EC involves more than simple online transactions, it encompasses diverse activities as conducting market research, identifying opportunities and partners, cultivating relationships with customers and suppliers, document exchange and customer modeling.

Our paper deals with the latter aspect of EC. We introduce here a distributed model for developing an EC adaptive system on the World Wide Web such that it can adapt its response to customers.

The model (presented for intranets in the companion paper [4]) is symmetric: we model both customers and goods and make both their profiles change as a consequence of a customer buying a certain product. The symmetry in our model greatly simplifies the approach and the queries, giving some insights on the

formalization of the allowed queries that were, in way, unexpected. Furthermore, the model itself can provide an easy-to-evaluate measure for the confidence in adapting its response to any given customer and is able to provide useful feedback to the manager, then allowing, so to speak, "manual adjustment" that can help the behaviour of the system in the future.

We present here the necessary extension to the model so that it can be used on a distributed system for EC on the World Wide Web. In fact, critical issue in the field of modeling customers and goods behaviour is to preserve the privacy of both the customer and the merchant. The former does not want that his behaviour can be used for purposes different from the "Helping serve you better" philosophy. The latter does not want that information on his transactions can be used by other merchants.

The issue of privacy is paramount in any EC distributed system that is to be deployed on the Internet. In fact, market research focuses on using information gathered about customers to refines sales efforts.

The solution presented is able to fulfill the requirements about the privacy and, at the same time, provide both customers and merchants with an efficient system that helps the customers in finding a "personalized" approach in the electronic store and the merchants in selling on a targeted and analyzed audience.

1.1 EC Systems and World Wide Web

In the last few years, many systems have been proposed and implemented that are meant to offer capabilities to support Electronic Commerce. Here, we try a first categorization of these systems by using a real-world analogy of a shop that has a Window, where potential customers can see goods, has a Clerk that is able to intelligently interact with customers, sell them goods and, finally, has a Cash where customers pay (either with cash, checks of cards).

Window Systems: in the early 90s, the WWW was recognized as a very effective way to promote products of private enterprises. These systems were essentially not more than an "*Electronic Window*" that can be browsed by each customer so that, essentially, he can browse the catalog of the producer without moving from home. Advantages of these systems over the more traditional ways, are that information can be promptly updated on the server side and, therefore, immediately available to customers.

Clerk Systems: Later, it was recognized that customers (once they have made up their mind) found awkward to place the order by ordinary mail, phone or fax. Then, systems allowed the customer to browse the catalog and, at the same time, proceed to shop with an *"electronic cart"*. Customers could buy items, put them back on the virtual shelfs and so on. Finally, the order was placed by using WWW and confirmed/checked through ordinary systems for the payment. As an example, many systems require a preliminary step when the customer (by fax or mail) must send in information (as Credit Card numbers) and, in response, the customer can obtains a unique ID to the system. The order, once placed, sometimes is checked toward a previously

given e-mail address and payments is, then, performed through ordinary ways. A well-known example is the protocol used until July 1997 by an electronic bookstore named Bookpool at `http://www.bookpool.com`.

Electronic Cash Systems: Much attention is given, nowadays, to these kind of systems, where payment is done entirely by digital means, by using wallets, cryptographic protocols that can guarantee both the sender and the receiver of the others' identity and guarantee the privacy of the communication by avoiding eavesdropping.

Our interest is on Clerk Systems that are receiving far less attention than Electronic Cash Systems. Our goal is to design a mechanism that allows for interactive, intelligent, guided and adaptive shopping that is certainly more fruitful both from the customer's perspective (finding interesting items in less time) and from the merchant's perspective (being able to respond to customer's unexpressed desires).

Although originally developed to facilitate information sharing (especially in academic environments), the World Wide Web has large potentials, partly already achieved, in the field of Electronic Commerce.

It is our opinion that in order to fully exploit WWW potentiality (user interface, portability, extensibility and widespread access) in the EC field, the merchant server must be a modified Web server, that must be aware of user's (the customer) behaviour so that it can take into account his interests and, as a consequence, provide the user with the right products, in few words it must be adaptive.

The problem, in a certain way, is similar to those experienced when WWW is used as an educational support, especially for distance learning. Several educational systems based on WWW have been developed in the recent past [8, 14] and some of them try to introduce adaptivity as a powerful tool to provide a better response, see [11] for a detailed bibliography.

In general, such adaptivity mechanism can be helpful to avoid the risk that users can "*have trouble in finding the information they need*" [16] because of the large amount of information available. An adaptive response from server can be able to avoid the "*information overloading*" by slowly increase the amount of information given to the user, information that is carefully chosen given the knowledge of the user.

1.2 The Privacy Issue in EC

Market research focuses on using information gathered about customers to refines sales efforts. The process of gathering, storing, purchasing or selling information about customers often violates the privacy rights of the customers, who may never have given permission for information to be used [10]. Europe's privacy regulations restrict the use of personal information far more than those in the United States, where marketing interests tend to dominate. The EU Privacy Directive in particular forbids collection of certain data about EU citizens, including union membership, race, religion, health information, sexual preference

and informations about the interactions that the customer has with a system of marketing.

This means that:

1. No personal data should be transmitted without adequate privacy protection;
2. Customer must be able to view and correct their data;
3. Companies must notify customers of the intended use for collected data;
4. Companies cannot release personal data to others without consent;
5. Companies cannot adjust owner organization respect to the customers interactions and cannot improve their sales e cannot personalize customers requests.

2 The model

In this section we describe the model in a simplified scenario where a single commercial server is available and several customers access the server. This scenario is for ease of description but is not totally unrealistic. The model in a distributed environment is described in the following sections.

We assume that customers are divided in categories, as well as items. When a new customer arrives, or a new item is available for selling, it is possible to assign it a *profile* corresponding to a stereotype that is chosen at the beginning. Stereotypes can be, of course, different and chosen as a consequence to the customer/item belonging to a category.

Informally, the idea of the profile of a customer is to define a category and a vector of values that indicate *his* current predisposition to buy goods from certain products categories. Symmetrically, an item of merchandise has a category and a vector of values that indicate *its* predisposition to be bought from a customer of each category. The symmetry here is an innovative aspect of the method: we can develop queries that go both ways and that, in our opinion, are a surprisingly formalization of common practise in the EC field.

We propose, here, also, several queries that can be performed on such a model and that can be used either to adapt the response toward customers but also to adjust the behaviour of the system by providing a useful feedback to the manager. For example, critical to the convergence of the system toward good responses is choosing right stereotypes and categories. Feedback can indicate that some stereotypes are too crude or that some categories have to be split in half or two categories joined.

We describe the requirements that are to be obtained by our model. This requirements are expressed in terms of easy and efficient management from the merchant side. The model:

- must model customer's behaviour: if a customer interacts with the system, the status of the system after the interaction should depend on the interaction;

- must be symmetric. Queries available for customer modeling should be also usable for merchandise modeling,to help the store manager;

- must be able to measure the *surprise* of a particular customer buying a certain item of merchandise in order to provide feedback to the store manager; i.e., this could trigger an *alarm* for the manager of the electronic store, maybe the prize is too low, or maybe it is time to buy more items of the same kind (since they are particularly appreciated by people of an unexpected category) or maybe the customers' stereotypes are surpassed by the society and it may suggest that some stereotypes are too crude or plainly wrong;

- must be able to provide a measure of the confidence of the mechanism in the forecast of the customer's behaviour;

- must be able to characterize *groups* of homogeneous customers or items. That is, we would like to get a sample of customers whose behaviour is similar enough, maybe in order to propose them a "special offer". Or, on the other side, we would like to group *close* items (i.e. for example, directed to the same audience) in such a way to lower prizes for them. Or, maybe even more important, start from a set of homogeneous customers, then observe them and propose to one of them what the other has recently bought.

At the same time, there are characteristics that we would not like in our model, such as follows:

- We do not want to keep history information: each customer can have some field where special offers are selected for them ad offered as soon as they get connected.

- We do not want computationally costly operations to be performed on the model: our system can have as many customers as millions and (for example) averaging over all the customer is simply unacceptable.

2.1 Queries in our model

In the rest of the paper, we call *Customer* any buyer that is connected to a commercial site (called *Merchant*).

The queries that we would like to be answered by our model are divided in two class: we show now queries in the interaction Customer/Merchant.

- *CUSTOMER: What is interesting to me?*
  List of the "top ten" products for a given customer (with a measure of confidence in the forecast).

- *MERCHANT: Who can be interested in this?*
  List of the "top ten" customers for a given item of merchandise (with a measure of confidence in the forecast).
- *CUSTOMER: Which merchandise group does this item belong to?*
  What are the goods that show a certain degree of affinity with this particular good I just bought (with a degree of confidence in the forecast).
- *MERCHANT: Which customers group does this customer belong to?*
  What are the customers that show a certain degree of affinity with the customer that just bought this good (with a degree of confidence in the forecast).

Now we describe the queries that are related to system management and marketing policies. They are meant to provide effective and efficient *feedback* to the management.

- *MERCHANT: Which confidence do I have in profile of this customer/good?*
  The merchant should provide a measure of behaviour of the system on customers/goods and of its confidence.
- *MERCHANT: Should I change the stereotypes?*
  if all (or a large percentage of) the customers of the same category move together toward a *limit* configuration, maybe this should be the stereotype. The same for items of merchandise.
- *MERCHANT: Should I change the categories?*
  If all the customers of a given category split in exactly two subcategories (with relative limit configurations) maybe I should split the categories as well. The same for items of merchandise.
  Moreover, it is easy to set an *alarm* that when the *surprise* of a transaction (the term is defined shortly) is above a threshold is able to asynchronously call the manager to inform him that there is something strange.

2.2 The theoretical model

As we said at the beginning of section 2, the profile of a customer consists of his category and a vector indicating the degree of interest (probability values) that the customer has with respect to each kind of merchandise. The same can be symmetrically said for goods' profiles. We now define formally the profiles and the model.

For x, y positive integers, let $A(x, y)$ be a set

$$A(x, y) \subseteq \{1, \ldots, x\} \times [0, 1]^y.$$

An item $\mathbf{a} \in A(x, y)$ is a pair $(c, \langle a_1, a_2, \ldots, a_y \rangle)$. The first component of $\mathbf{a}$, denoted by $\mathrm{Cat}(\mathbf{a})$, is called the *category* while the remaining y-item vector is called its *configuration*. A configuration is *legal* if $\sum_{i=1}^{y} a_i = 1$.

The categories implicitly define a partition of $A(x, y)$ in x subsets $A_1(x, y)$, $A_2(x, y)$, ..., $A_x(x, y)$ where

$$A_c(x, y) = \{\mathbf{a} \in A(x, y) \mid \mathrm{Cat}(\mathbf{a}) = c\}.$$

The *symmetric adaptive* model is composed by two sets:
• the set of customers $U = A(d, k)$ where customers are partitioned in *divisions* $U_1, U_2, \ldots, U_d$;
• the set of merchandise $M = A(k, d)$ where items are partitioned in *kinds* $M_1, M_2, \ldots, M_k$.
A customer $\mathbf{u} \in U$ is, therefore, characterized by his *division* $\mathrm{Cat}(\mathbf{u})$ and by a configuration vector that has as many components as the kinds of merchandise available. Analogously, a merchandise item $\mathbf{m} \in M$ has its own *kind* $\mathrm{Cat}(\mathbf{m})$ and its configuration vector has as many components as the number of divisions.

Let us define the *surprise* of a customer $\mathbf{U} = (c, \langle u_1, \ldots u_k \rangle)$ related[1] to an item of merchandise $\mathbf{M} = (s, \langle m_1, \ldots, m_d \rangle)$ as the function

$$S(\mathbf{U}, \mathbf{M}) = 1 - u_s m_c$$

where u_s and m_c represent the elements' values of the profiles of user and good that are related.

This function measures the information that is given to the system when $\mathbf{u}$ buys $\mathbf{m}$ and goes from 0 (absolutely expected) to 1 (maximum surprise).

The function *distance* is the usual euclidean distance between points in $\{0,1\}^k$ for customers and in $\{0,1\}^d$ for items of merchandise.

2.3 The operations

• *Buying a product and updating profiles.*

What happens to our model when customer $\mathbf{U} = (c, \langle u_1, \ldots, u_k \rangle)$ wants to buy item $\mathbf{M} = (s, \langle m_1, \ldots, m_d \rangle)$? Let

$$P = \max\{S(\mathbf{U}, \mathbf{M}), u_s\} \quad \textit{and} \quad Q = \max\{S(\mathbf{U}, \mathbf{M}), m_c\}.$$

Then, in $\mathbf{U}$ the value P takes the place of u_s and in $\mathbf{M}$ the value Q takes the place of m_c and this means that either configurations do not change if the event is not a surprise, that is $S(\mathbf{U}, \mathbf{M}) \leq u_s$ or $S(\mathbf{U}, \mathbf{M}) \leq m_c$. If a profile changes due to a surprise, it has to be made legal (i.e. summing up to 1) and, therefore, assuming that $U' = (c, \langle u'_1, \ldots, u'_k \rangle)$ and $M' = (s, \langle m'_1, \ldots, m'_d \rangle)$ are, respectively, the customer and good profiles after $U = (c, \langle u_1, \ldots, u_k \rangle)$ has bought the item of merchandise $M = (s, \langle m_1, \ldots, , m_d \rangle)$ we have that:

$$u_i = \begin{cases} \frac{1 - u_s \cdot m_c}{C} & \text{if } i = s \\ \frac{u_i}{C} & \text{otherwise} \end{cases}$$

$$m'_i = \begin{cases} \frac{1 - u_s \cdot m_c}{D} & \text{if } i = c \\ \frac{m_i}{D} & \text{otherwise} \end{cases}$$

where $C = \sum_{i \neq s} u_i + (1 - u_s m_c)$ and $D = \sum_{i \neq c} m_i + (1 - u_s m_c)$

[1] We want to emphasize by the term "related to" the bidirectionality of the system. We do not say that $\mathbf{U}$ *buys* $\mathbf{M}$.

Notice also that, the preference of a certain customer U for items of kind i, given by u_i, can be increased due to the surprise of U buying an item of kind i that is not usually sold to people of the same category of U. The value u_i can be also decreased due to surprises in other fields that make necessary make the configuration legal (i.e. summing up to 1).

- *"Top 10" queries.*

Here we describe the "Top 10" query for a customer, the analog query for items of merchandise is totally symmetric.

Assume we want to present T items of interest for customer $\mathbf{U} = (c, \langle u_1, \ldots, u_k \rangle)$. Such a list is built with the following proportions: there are as many as $u_i \cdot T$ items[2] for each kinds of goods i.

What are the goods in kind i selected for "top T" products for customer **U**? We select the items that are most wanted by customers in category c, that is we pick the goods that have maximum m_c.

We emphasize here the advantages of the symmetric model: queries are easily obtained by reverting things in the model.

- *Affinity queries.*

Because of space limitation, we sketch here how affinity queries can be performed in our system. Recall that the second component of customer profile is a vector with k elements. The distance function used is the usual euclidean distance in the sphere S with radius 1 and center t the origin in the space $\{0,1\}^k$. Now, an affinity query can be easily seen as a query of the kind: given a point U in sphere S, find all the points within distance ℓ from U. The same is done for items of merchandise in $\{0,1\}^d$.

2.4 Confidence of the system

In this subsection, we sketch how our model can be used to give a measure of degree of confidence in the operations previously described. More details can be found in [4].

The confidence operations can be used to provide useful feedback to the manager and consists in evaluating the confidence that can be assumed for customers' profiles. The confidence for goods' profiles can be easily obtained by clerical changes, since the symmetry of our model.

Let us define the entropy function of a customer U and of an item of merchandise, that are random variables respectively as follows[3]:

$$H(U) = -\sum_{i=1}^{k} u_i \log u_i \quad \textit{and} \quad H(M) = -\sum_{i=1}^{d} m_i \log m_i .$$

In the sequel, we give a very brief definition and interpretation of the entropy as defined by C.Shannon [19]. The interested reader may consult [2]. Entropy

[2] At this moment we are disregarding rounding problems. Solutions *"ad hoc"* can be easily adopted.

[3] All the logarithms in the rest of the paper are base 2

function is meant to measure the average uncertainty of a random variable X. $H(X)$ does not depend on the values the random variable assumes, but only on probabilities associates with those values. In our model, U represents the probability distribution of the customer buying products of a certain category. The same interpretation can be given of the item of merchandise M.

The entropy of a random variable with ℓ values ranges from 0 to $\log \ell$. The smaller the values of $H(X)$ are, the more certain is the outcome of the random variable X. In our model, if entropy of a customer U approaches to $\log k$ then it means that customer U is, more or less, buying items from all the kinds.

It is possible to consider the relationship between the interactions number of a customer and her entropy and have, therefore, a partition of customers in subsets of "well-behaving" and (hopefully) correctly modeled since their interactions are considered sufficient by the system and subsets of customers that (for different reasons) are not as well modeled by the system.

How do we use the confidence in the queries previously defined? Every time a customer U is given back as a result of the query, the system sends the confidence along with the results. The confidence can be used as a numerical measure to guide into the interpretation of the results. For example, consider a low confidence for a customer given back from a "Top 10 customers" query. Recall that this query is needed, for example, when a new item is arriving into the system and the manager wants to find the best potential customers[4]. Now, a low confidence given to a customer U means for the manager, that, for example, customer U had only few interactions with the system and the fact that he is in the "Top 10 customer" for the given item may depend on his stereotype (the one assigned by the system at the entrance) and not by his behaviour that is not well known by the system.

3 The model in an untrustworthy environment

The model we just described is strongly based on the assumptions (true on intranets) that customers buy from a unique, single commercial server which is often unrealistic in the real world. These assumptions are further limiting the possibility that customers exports their profiles toward several distinct commercial servers, often competing with each other.

In this section, we first briefly review the off-the-shelf protocols used to ensure privacy on the Internet nowadays (based on symmetric and asymmetric keys) and then present the architecture design to overcome these problems that arise from an untrustworthy environment.

3.1 Security and authentication

In our architecture, we use some encryption algorithms in a communication scheme composed of symmetric (DES) and asymmetric key (RSA) algorithms,

[4] It can be also used for special offers.

that ensure the privacy of participants (overcoming an eventual malicious listener on the network) and the authentication.

Symmetric key algorithms Symmetric key algorithms are used for the bulk encryption of data or data streams. These algorithms are designed to be very fast and (usually) have a large number of possible keys. The best symmetric key algorithms offer near-perfect secrecy: once data is encrypted with a given key, there is "no way" to decrypt the data without possessing the same key. We use DES (Data Encryption Standard) that is a block cipher that uses a 56-bit key and has several different operating modes depending on the purpose for which it is employed.

Asymmetric key algorithms. Public-Key cryptography, also known as asymmetric cryptography, uses two keys: one key to encrypt the message and the other key to decrypt the message. The two keys are mathematically related such that data encrypted with either key can only be decrypted using the other. Each user has two keys: a public key and a private key. The user distributes the public key. Because of the relationship between the two keys, the user and anyone receiving the public key can be assured that data encrypted with the public key and sent to the user can only be decrypted by the user using the private key. This assurance is only maintained if the user ensures that the private key is not disclosed to another. Therefore, the key pair should be generated by the user. The best known public-key cryptography algorithm is RSA (named after its inventors Rivest, Shamir and Adleman).

Confidentiality is ensured by the use of message encryption. When two users want to exchange messages securely, each transmits one component of their key pair, designated the public key, to the other and keeps secret the other component, designated the private key. Because messages encrypted with the public key can only be decrypted using the private key, these messages can be transmitted over an insecure network without fear that an eavesdropper can use the key to read encrypted transmissions.

Integrity and authentication are ensured by the use of digital signatures. Because of the mathematical relationship between the public and private keys, data encrypted with either key can only be decrypted with the other. This allows the sender of a message to encrypt it using the sender's private key. Any recipient can determine that the message came from the sender by decrypting the message using the sender's public key.

Integrity is ensured by use of message digest. A message digest is a value generated for a message (or document) that is unique to that message. A message digest is generated by passing the message through a one-way cryptographic function, i.e., one that cannot be reversed. When the digest of a message is encrypted using the sender's private key and is appended to the original message, the result is known as the digital signature of the message. The recipient of the digital signature can be sure that the message really came from the sender by decrypting the message using the sender's public key. And, because changing

even one character in the message changes the message digest in an unpredictable way, the recipient can be sure that the message was not changed after the message digest was generated.

3.2 The architecture

The scenario we envisage is the following: there are several commercial servers, and several customer communities that want to access and, possibly, buy goods from the commercial servers using synthetic information about their behaviour stored in profiles.

The issue here is privacy: in such a distributed environment, the system should respect privacy of everybody. By ensuring the privacy of both customers and merchants we must guarantee the following two requirements:

1. We **must** ensure that analytical information about sold goods to customers from a certain merchant is kept hidden from other merchants.
2. We **must** ensure customers privacy in a twofold way: on one hand, analytical information about transactions with merchant X is known only to X; on the other hand customer profiles should be kept without any reference to customer's identity.

The architecture of our model in an untrustworthy environment is based on three components that communicate under a strong authentication scheme (briefly described below):

1. A Profiling Authority (PA), that holds the customers' profiles, distributes a customer profile to the merchant servers and receives from the merchant the transactions list (containing only goods' profiles) and updates customer's profile.
2. A Merchant, that is autorized to ask customers profiles to the PA so it can use them to improve its response toward customers satisfactions. Of course, it still keeps its local historical information on goods sold to each customer as well as the profile of its items of merchandise. When the customer buys some goods, there is a protocol (described below) to update customer and goods' profiles.
3. A customer, that has to be enrolled at the PA, can allow Merchants to use its profile (once per transactions) to adapt responses. At each interaction in the work session with a Merchant the customer enables the merchant to use the profile using a protocol described below.

Notice that customers profiles are synthetic information about life-long transactions with all the merchants. Therefore, we do not break customers' privacy if a profile is given to a single merchant, since the merchant can only try to match it against its own local information about its transactions with that customer. On the other hand, information about customers' identity is hold on merchants (because of payment information) and is kept hidden from the PA by using an anonymous customer ID. Finally, customers' profiles loose meaning as time

passes by, and therefore, it would be useless for the merchant to try to collect profiles.

One of the other advantages of our architecture design is that it makes meaningless for the merchant to hold local customers statistics because the profile is built upon all the transactions and is much more significant.

Crucial to the privacy is the scheme used to *(a)* distribute customers' profiles to merchants and *(b)* authenticate the goods' profiles sent from merchant to PA We only sketch here the schemes because of space limitations.

As we said before, the architecture of our model is based on three components that must communicate in a secure way. Each participant uses a scheme of asymmetric cryptography to send message to each other. PA uses DES (symmetric key algorithm) to encrypt and decrypt the key to access to the profile. The DES key and the associated customer ID are passed among the participants each time.

In the rest of this section, we briefly describe the scheme of the operations for managing profiles in a distributed environment. We emphasize that the operations, as sketched here, are designed for a client-server approach, like the WWW, and therefore, can be easily implemented by using standard browsers (Netscape, Microsoft Internet Explorer) with additional security packages.

Notice that, also when not explicitly said, all messages that are exchanged among participants are encrypted with public key's receiver and signed by sender.

Enrolling a new customer to the PA

1. The customer gets the public key of the PA and requests a new profile enclosing his own public key.
2. The PA, according to some previously agreed upon criteria, chooses among available stereotypes and assigns it to a customer ID.
3. PA generates a DES key (symmetric cryptography).
4. PA holds in its archives customer ID, his public key, the DES key and the profile.
5. PA sends to the customer his customer ID and the DES key encrypted with his public key.
6. The customer decrypts customer ID and DES key with his private key.

Customer requests a personalized service to the merchant. Among the personalized services we enclose all the queries that, in order to be performed by the Merchant, need the customer profile.

1. The customer sends his ID and the DES key encrypted with the PA's public key (and with the customer digital signature) to the Merchant.
2. Merchant can forward the request to the PA in order to get the profile.
3. PA generates a new DES key.
4. PA send back a message to the Merchant consisting of two pieces:
   a) customer ID and (new) DES key encrypted with customer's public key with the PA's digital signature;

b) customer profile encrypted with the Merchant public key.
5. The Merchant can now read customer profile and build the personalized answer for him.
6. The Merchant sends back to the Customer the result of the query and the first part of the message received by the PA, containing the new DES key, encrypted with customer public key.

Updating the Profiles

1. The customer sends his ID and the DES key encrypted with the PA's public key (and with the customer digital signature) to the Merchant and a request of purchase.
2. Merchant sends a message to the PA consisting of two pieces: customer ID and DES received, and the list of goods' profiles encrypted with the public key 's PA (and with digital signature).
3. PA updates customer profile based on goods's profiles and generates a new DES key for the new profile.
4. PA send back a message to the Merchant consisting of two pieces:
   a) customer ID and (new) DES key encrypted with customer's public key with the PA's digital signature;
   b) Acknowledge encrypted with the Merchant public key.
5. The Merchant sends back to the Customer the first part of the message received by the PA encrypted with customer public key and his acknowledge. At the same time, the Merchant can update its goods' profiles.

3.3 The queries in the distributed environment

Obviously, queries are changed in the distributed environment because of the (necessary) asimmetry that is introduced in the model.

Therefore, there are some queries that, now, lack of the simmetry of the trustworthy environment. For example, the affinity query for the goods can be done only by a Merchant and, of course, only on its own data, while the affinity queries on the customers can be done only by the PA.

Such asimmetry can be easily solved by "ad hoc" queries. For example, it is possible to envison a scenario where the Merchant is able to ask the PA to perform affinity queries on the customers profiles or that the PA is sending the confidence values on customers profile each time the profile is required by some Merchant.

4 Conclusions and Further research

We mentioned in the introduction that our model is, at the same time, simple enough to be easily described, implemented, tested and evaluated and, on the other hand, it is opening new avenues for more refined kind of queries that can be performed and analized.

The "intranet" version of the model (no PA, only one merchant server and several customers) is easy to implement and can be helpful for unsuspected uses. During a fruitful collaboration with *Italdata s.p.a*, a prototype of a local server implementing the model has been developed. The prototype (*EcoNET*) is based on Microsoft Merchant Server [15] and is able to coordinate the buying process internal to a large enterprise. Each employee has his own profile and is presented with products that can be interesting for him.

Our model in the distributed framework is designed so that it can be easily implemented over the existing WWW infrastructure, being based on client-server queries. It is not difficult to modify existing authentication-enabled browsers so that they can deal with "profiles" rather than certificates. The modifications to servers (both Merchant and Profiling Authority) appear even more straightforward.

One of the field where we feel there is much room for improvement is on the feedback to the manager section. An accurate statistical analysis on the field can decide to develop more refined *trigger* mechanisms, such as deciding to discard (as statistical not relevant) exceptionally high values of surprise.

We see this as one of the greatest advantages of the model (we should say of *any* model): it is easy enough to design a system on it and, at the same time, can be easily enlarged with "*ad hoc*" mechanisms to real-world applications.

Acknowledgments. We gratefully thank Prof. Alfredo De Santis for helpful discussions. We thank Italdata s.p.a. for the collaboration in the development of the intranet-based prototype of EcoNET and, particularly, Dott. Giuliano Calabrese for his cooperation. Work of Maria Barra was partially supported by a grant by SESM - *Sistemi Esperti per la Manutenzione*, Napoli (Italy).

References

1. M. Abadi and R. Needham. "*Prudent engineering practice for cryptographic protocols*". 1994.
2. R.B.Ash. "*Information Theory*". Dover Publications,Inc. New York 1965.
3. D.Denning "*Cryptography and Data Security*". Addison-Wesley Publ., 1983
4. M.Barra, G.Cattaneo, A.Negro, V.Scarano. "*Symmetric Adaptive Customer Modeling in an Electronic Store*". Submitted for publication, 1997.
5. T. Berners-Lee. "*World Wide Web Initiative*". WWW Home Page. [URL: `http://info.cern.ch/hypertext/WWW/TheProject.html`]
6. T. Berners-Lee, R. Cailliau, J.F.Groff. "*The World Wide Web*". Computer Networks and ISDN Systems, Nov. 1992, vol.25 (no.4-5), 454-9.
7. T. Berners-Lee. "*Hypertext Transfer Protocol*". Internet Draft.
8. D. Dwyer, K. Barbieri, H.M. Doerr. "*Creating a Virtual Classroom for Interactive Education on the Web*". Proc. of WWW 95, Third Int. Conf. on World Wide Web.
9. V. Bush. "*Memex Revisited*". In *Science is not enough*, W.Morrow and Co. Reprinted in Nyce, J.M. and Kahn, P. (Eds.) (1991), From to Hypertext: Vannevar Bush and the Mind's Machine. Academic Press.
10. D.Cameron. "*Electronic Commerce. The New Business Platform for the Internet* ". Computer Tecnology Research Corp. 1997.

11. S. Ferrandino, A. Negro, V. Scarano. "CHEOPS: Adaptive Hypermedia on World Wide Web". Proceedings of the European Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services (IDMS '97), 10-12 Sett. 1997. Ed. Springer-Verlag (LNCS).
12. R.Fielding, H. Frystyck, T. Berners-Lee. "*Hypertext Transfer Protocol,* HTTP *1.1*". HTTP Working Group Internet Draft.
13. N. Hammond, L. Allison. "*Extending Hypertext for learning: An investigation of access and guidance tools*". In Sutcliffe,A. and Macaulay, L. (Eds.): *People and Computers*, Cambridge University Press, 1989.
14. B. Ibrahim, S.D. Franklin. "*Advanced Educational Uses of the World Wide Web*". Proc. of WWW95, 3rd International Conference on World Wide Web.
15. Microsoft Corporation. "*Merchant Server on-line documentation*". from CD-ROM of Merchant Server 1.0 version.
16. J. Nielsen. "*Hypertext and Hypermedia*". Academic Press Ltd, 1990.
17. R. Rivest. RFC 1319: The MD2 Message Digest Algorithm. April 1992.
18. B. Schneier. "*Applied Cryptography: Protocols, Algorithms, and Source Code in C*", Published by John Wiley & Sons, Inc. 1994.
19. C.E. Shannon. "*A Mathematical Theory of Communication*", *Bell System Tech. J.*, **27**, pp.379-423, 623-656.