Customer Information Sharing between E-commerce Applications

Liliana Ardissono, Marco Botta
Luca Di Costa, Giovanna Petrone
Dipartimento di Informatica
Università di Torino
Corso Svizzera 185, Torino, Italy
Email: {liliana, botta, giovanna}@di.unito.it

Fabio Bellifemine, Angelo Difino
Barbara Negro
Telecom Italia Lab
Multimedia Division
Via Reiss Romoli, 274 - Torino, Italy
Email: {Fabio.Bellifemine, Angelo.Difino,
Barbara.Negro}@tilab.com

***Abstract*— The management of one-to-one business interaction is challenged by the latency in the acquisition of information about the individual customer's preferences. Although sharing this type of information would empower service providers to personalize the interaction with new customers since the first connection, this idea can be hardly applied in real cases if the service provider cannot protect the data it has acquired from competitors and select the trusted parties from which it wants to receive information.**

As a solution, we propose a framework supporting the controlled sharing of customer information between e-commerce applications. Our framework includes two main components: 1) a Trust Management System (running off-line with respect to the information sharing service), which enables the service provider administrator to specify restrictions on the service providers to be considered as trusted parties; 2) a User Modeling Agent, which manages the propagation of customer data between service providers, given their trust relationships. The User Modeling Agent also takes care of combining the customer information provided by the trusted parties in order to generate an overall view of the customer preferences.

I. Introduction

Various techniques have been applied in Web-based stores and electronic catalogs to personalize the recommendation of products; see [1], [2], [3], [4]. For instance, collaborative filtering [5] steers the recommendation of goods by analyzing the similarities in the purchase histories of different people. Moreover, content-based filtering (e.g., see [6]) recommends goods having properties that the individual customer preferred in the past. In all cases, the customer's behavior has to be observed for some time in order to acquire a user model describing her preferences. Thus, a delay occurs before the service provider application personalizes the interaction in an effective way.

Indeed, the preference acquisition process can be speeded up if the service providers exchange their customer information with one another. For instance, if two book sellers trust each other, they might share the user models describing their customers in order to increase the knowledge about the common customers and to extend the set of visitors they can handle as known ones. In Business to Customer e-commerce, several service providers already exploit their own user modeling systems to analyze clickstream data and locally manage their customers' profiles. For these providers, the main purpose of sharing customer information with other (trusted) parties is that of acquiring information about unknown customers (first time visitors) or recently acquired customers, whose profiles are not yet complete.

In this paper, we propose a framework supporting a controlled propagation of customer information among e-commerce applications. The framework includes a Trust Management System that enables the administrators of individual service providers to specify their trust relationships with other providers and to examine the set of service providers eligible for information sharing, possibly modifying it by adding and removing individual service providers. Moreover, the framework includes a User Modeling Agent that coordinates the exchange of customer information according to the network of declared trust relationships: when a service provider requests information about a customer, the User Modeling Agent merges the information provided by the trusted parties into a user model ready to be exploited for personalization purposes.

From the viewpoint of trust management, our framework enables the service provider administrator to select partners for information sharing both at the individual level and at the class level (on the basis of their features). More generally, our framework has the following advantages:

- Service providers are supported in the information sharing by a trusted third party (the User Modeling Agent).
- Service providers do not need to modify the core of their applications when they register for information sharing. In fact, each application may continue to exploit its own personalization system: the application may personalize the interaction with an individual customer by exploiting its local user model, the model provided by the User Modeling Agent, or it may integrate the two models.
- A service provider not equipped with its own user modeling system may question the central User Modeling Agent when it needs information about a customer and exploit the returned information for personalization purposes.

In this paper, we will focus on the Trust Management System, which provides the basis for the customer information propagation, and we will only sketch the main aspects of the User

M. Baldoni, F. De Paoli, A. Martelli, and A. Omicini (Eds.): *WOA 2004 – Dagli Oggetti agli Agenti, Sistemi Complessi e Agenti Razionali*, Torino, Italy, November 29th – December 1st 2004. Pitagora Editrice Bologna. ISBN 88-371-1533-4. http://woa04.unito.it

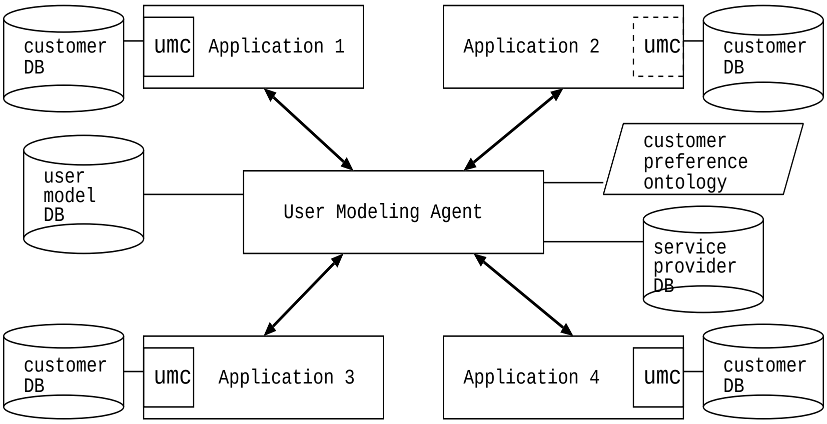


Fig. 1. Framework Architecture.

Modeling Agent. The rest of this presentation is organized as follows: Section II outlines some basic issues concerning customer information sharing between heterogeneous applications. Section III describes the architecture of our framework and the regulation of the propagation of information from service provider to service provider. Section IV describes the management of trust relationships between service providers. Section V compares our proposal to related work. Section VI discusses possible extensions to our work and closes the paper.

II. Background

In the development of a service supporting customer information sharing between applications the following issues are relevant:

1) In the propagation of the information, privacy preferences have to be taken into account [7], [8]. For instance, a customer might want to make her personal data available only to the service providers she is interacting with, she might allow the propagation to providers belonging to restricted categories, such as book sellers, or she might restrict the propagation of her personal data to service providers conforming to privacy policies [9].
2) Ontology mapping issues have to be addressed in order to enable the propagation of information in an open environment.
3) The information collected by each application has to be propagated to other applications according to specific trust relationships. For instance, a service provider may impose accessibility restrictions on the information it collects, or it may be interested in receiving information from selected sources. For instance, a book seller might want to share information only with other book sellers and to ignore data acquired by music sellers. Moreover, it might not want to share information with some particularly untrusted book sellers.

The issues described in the first two items above are addressed in initiatives that are proposing standard solutions to be adopted by the applications. For instance, the Platform for Privacy Protection initiative of the W3C Consortium (P3P, [9]) is defining a standard representation language for the specification of privacy preferences and privacy management policies. The ultimate goal is to enable the specification of privacy preferences at the customer side (e.g., in the user agent of a Web browser) and the automated verification of the acceptability of the policies adopted by the web sites the user is visiting.

As far as the binding task is concerned (item 2 above), this is very a complex issue and has usually been addressed by adopting ad hoc solutions. However, the current attempts to solve this issue tend to propose standard ontologies for the representation of user information and preferences, with the goal to make applications exploit a uniform representation language for the description of their users. Specifically, in the P3P proposal, a user information ontology has been defined to describe basic customer data such as contact addresses, socio-demographic information and clickstream data. Moreover, in order to enable service providers to declare the kind of user preference they want to collect, the ontology can be extended with additional concepts. This means that standard preference ontologies can be developed for the main sales domains, similar to the representation of products in the RosettaNet initiative [10]. Furthermore, in the research about Semantic Web, complex ontologies are being proposed to represent rich user preference information; e.g., see [11], [12].

In the rest of this paper, we will focus on the third issue, which has been relatively unexplored. For simplicity, we will describe the user preferences in a trivial <feature, value> representation language, as the focus of this presentation is in the controlled propagation of information, not on the kind of exchanged data. Moreover, given the trend towards the standardization of ontologies, we will assume that the User Modeling Agent adopts a general ontology and that the applications registered for information sharing adopt a subset of that ontology, without handling ontology mapping issues. Finally, we will assume that customers do not impose any restrictions on the propagation of their personal data although we believe that our framework can be extended to manage privacy preferences by conforming to the P3P

```
Customer preferences:
  Books:
    history: (Int:[0,1], Confidence:[0,1]);
    science: (Int:[0,1], Confidence:[0,1]);
    scienceFiction: (Int:[0,1],Confidence:[0,1]);
    literature: (Int:[0,1], Confidence:[0,1]);
    ...
  Music:
    rock: (Int:[0,1], Confidence:[0,1]);
    jazz: (Int:[0,1], Confidence:[0,1]);
    disco: (Int:[0,1], Confidence:[0,1]);
    ...
```

Fig. 2. Portion of the Customer Preference Ontology.

platform specifications without major problems.

III. Architecture of our Customer Information Sharing Framework

Before describing the Trust Management System, it is worth sketching the architecture of the customer information sharing service which controls the propagation of information between registered service providers.

We have designed a User Modeling Agent devoted to coordinating the propagation of information between service providers, by taking their mutual trust relationships into account. The architecture supports the cooperation between heterogeneous applications, that may (may not) exploit a local user modeling component for the management of the customer profiles. Our User Modeling Agent is also responsible for reconciling the information provided by the service providers, by merging the alternative user models in order to generate the preference information needed by each individual application.

Figure 1 shows the high-level architecture of our framework. The figure shows a scenario where four service providers have registered for information sharing. Each application has a local database (*customer DB*) storing its own customer information and may exploit a local user modeling component (*umc*) to manage the user models. The local user modeling component is shown as a small box within the application rectangle; in the example, three applications have their own component (plain boxes), while one application (*application 2*) only exploits the preference information provided by the User Modeling Agent (dashed box).

The *customer preference ontology* defines the representation of preferences adopted in the User Modeling Agent and in the registered applications. As the User Modeling Agent must be able to integrate the information retrieved from different service providers, the ontology is organized in subparts describing the customer preferences in different domains, e.g., the sales of books, music, and services such as insurance agencies. Figure 2 shows a portion of the ontology related to the books and music sales domains. For each concept:

- The preference is represented as an interest degree that takes real values in [0, 1]. The 0 value denotes total lack of interest, while 1 denotes maximum interest.
- The confidence degree describes the reliability of the estimated interest: it is a real number in [0, 1], where 0 denotes total lack of confidence (no evidence about the preference is available) and 1 denotes absolute confidence in the estimation.
  The confidence degree enables the User Modeling Agent to correctly integrate the information provided by the applications. In fact, each application is likely to provide evidence about few user preferences, leaving the other unknown and this is represented by setting the confidence to 0.

Similar to the approach adopted in other application domains (e.g., TV recommenders [13], [14]), the ontology is organized in a hierarchical way, as a tree of concepts, which supports a rather straighforward propagation of the interest and confidence information between concepts.

At the core of the architecture, the *User Modeling Agent* manages the registered applications. The agent exploits a *service provider DB* storing information about the applications registered for customer information sharing. As described in the following section, a registration service (the Trust Management System) enables service providers to join the set of registered applications and to specify trust relationships with the other service providers. The User Modeling Agent exploits these relationships to constrain the propagation of customer information within the pool of registered applications.

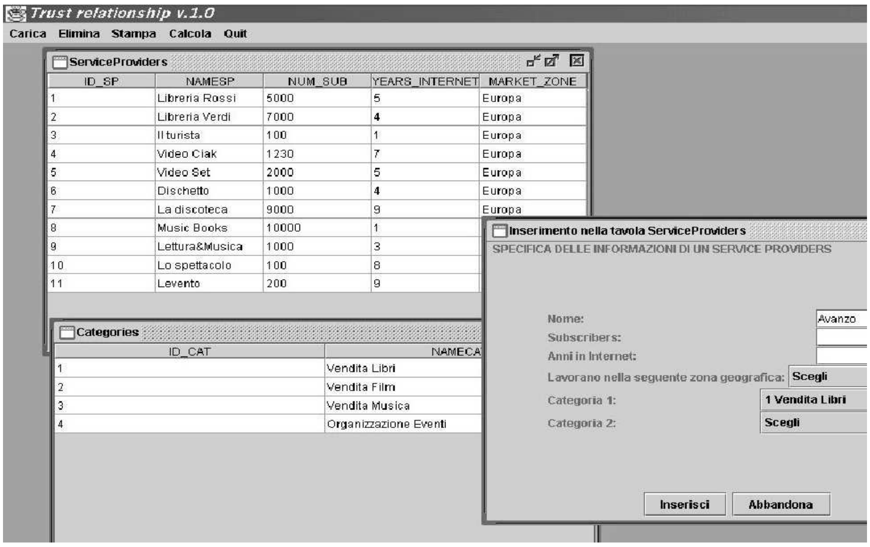
When an application SP invokes the User Modeling Agent to acquire the preferences of a customer C, the agent selects the trusted applications and requests C's user models. Then, the agent synthesizes the customer preferences, it generates the user model including the information needed by SP and sends the information to SP. The exchange of data between service providers and User Modeling Agent is carried out by means of SOAP messages storing the user models.

In order to merge the user preferences collected by different providers, the identifiers selected by the customer when registering for the various services have to be related to one another. As the identification of the customer across different applications is a very complex issue, global identifiers have been proposed, e.g., in the Microsoft Liberty Alliance project [15]. In our work, we adopt the global identifier approach: the User Modeling Agent maintains a central *user model DB* storing, for each customer, the identification data she entered when she registered at a service provider's site. In the absence of a global identifier (e.g., for customers who did not accept a global passport and who registered for a service before it registered for information sharing), multiple identities are treated as different customers.

IV. Trust Management

The trust relationships are specified by the service provider administrators when they register for customer information sharing and are stored in the *service provider DB* managed by the Trust Management System and exploited by the User Modeling Agent at information propagation time.

Similar to policy-based approaches [16], we adopted a concise and declarative representation of trust relationships based on the specification of service provider features and of



ServiceProviders

ID_SP	NAMESP	NUM_SUB	YEARS_INTERNET	MARKET_ZONE
1	Libreria Rossi	5000	5	Europa
2	Libreria Verdi	7000	4	Europa
3	Il turista	100	1	Europa
4	Video Ciak	1230	7	Europa
5	Video Set	2000	5	Europa
6	Dischetto	1000	4	Europa
7	La discoteca	9000	9	Europa
8	Music Books	10000	1	
9	Lettura&Musica	1000	3	
10	Lo spettacolo	100	8	
11	Levento	200	9	

Categories

ID_CAT	NAMECA
1	Vendita Libri
2	Vendita Film
3	Vendita Musica
4	Organizzazione Eventi

Inserimento nella tavola ServiceProviders

SPECIFICA DELLE INFORMAZIONI DI UN SERVICE PROVIDERS

Nome: Avanzo
Subscribers:
Anni in Internet:
Lavorano nella seguente zona geografica: Scegli
Categoria 1: 1 Vendita Libri
Categoria 2: Scegli

Fig. 4. Trust Management System: Introduction of Information about a Service Provider.

conditions on the propagation of data. However, we adopted an explicit trust management technique, based on the analysis of trusted party lists, instead of automatically providing access certificates. The reason is that, in an open e-commerce environment, the set of service providers having the right to receive the information collected by a service provider cannot be defined by means of necessary and sufficient conditions. More specifically, restriction conditions can be defined to select groups of entities eligible for information sharing. However, a one-by-one analysis is needed to revise the groups according to the requirements of the individual service provider, who may want to exclude candidates for business purposes. Notice that the evaluation of trust at the instance level is important not only because the customer information is very precious, but also because its dissemination is regulated by severe privacy rules that make both the service provider (as collector of personal data) and the middle agent(s) supporting information sharing responsible for any misuse of such data. Thus, each service provider administrator must be enabled to inspect and modify (by overriding general feature-based trust relationships) the list of parties to which the information sharing framework propagates the data.

```
Identification data:
   ID: SP1;
   Name: BookLand;
   URL: http://www.bookLand.com;
   ...
Categorization: bookSeller;
Features:
   NumberOfSubscribers: 3000;
   ...
Trust relationships:
   TAKE: {(bookSeller OR movieSeller) AND
         nrOfSubscribers>1000, 1), ...}
   NOT-TAKE: {musicSeller, ...}
   GIVE: {bookSeller OR movieSeller, ...}
   NOT-GIVE: {insuranceAgent, ... }
```

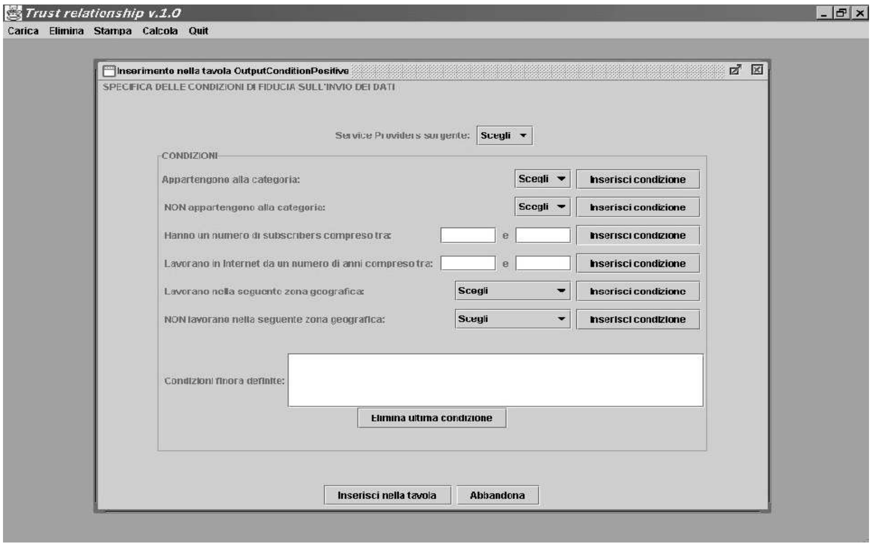
Fig. 3. Sample Service Provider Descriptor.

A. Description of Service Providers

Each service provider is described by the following data, stored in a table of the *service provider DB* (see Figure 3 for a sample descriptor):

1) Identification data: name, address, social security number, . . .
2) Categorization: each service provider is classified in one or more categories. A taxonomy specifies the service provider types handled by the User Modeling Agent; e.g., *bookSeller*, *musicSeller* and *insuranceAgent*.
3) Features: number of subscribers, Quality of Service, . . .
4) Trust relationships. These relationships are stored in separate fields, each one including one or more (alternative) relationships, separated by commas:

- *TAKE*: Conditions for the selection of the applications from which the service provider wants to receive customer information and degree of trust in the information.
  - The conditions are well-formed boolean expressions and may include categories and restrictions on the values of the service provider features.
  - The degree of trust is a real value in (0,1]. The value 1 denotes absolute trust, while values near to 0 denote lack of trust, i.e., the provider ignores the information coming from those providers.
- *NOT-TAKE*: Conditions for the selection of the applications from which the service provider does not want to receive information. These conditions have the same format as the previous ones but the trust degree is omitted because it is by default equal to 0.
- *GIVE*: Conditions imposed by the service provider on the dissemination of customer information to other service providers. These conditions are well-formed boolean


Trust relationship v.1.0
Carica Elimina Stampa Calcola Quit
Inserimento nella tavola OutputConditionPositive
Inserisci condizione
Elimina ultima condizione
Inserisci nella tavola
Abbandona

SPECIFICA DELLE CONDIZIONI DI FIDUCIA SULL'INVIO DEI DATI

Service Providers sorgente: Scegli

CONDIZIONI

Appartengono alla categoria: Scegli

NON appartengono alla categoria: Scegli

Hanno un numero di subscribers compreso tra: e

Lavorano in Internet da un numero di anni compreso tra: e

Lavorano nella seguente zona geografica: Scegli

NON lavorano nella seguente zona geografica: Scegli

Condizioni finora definite:

Fig. 5. Trust Management System: Definition of Trust Relationships.

expressions and may include categories and restrictions on the values of the service provider features.

- *NOT-GIVE*: Conditions for the selection of applications to which the service provider does not want to deliver its own customer information. The conditions have the same format as the *GIVE* ones.

Notice that by defining *TAKE* and *NOT-TAKE* relationships, the service provider assesses the usefulness and the quality of the preference estimates that might be provided by the other applications. For instance, the *TAKE* field of the descriptor of Figure 3 specifies that the *BookLand* service provider only trusts the information provided by book sellers and movie sellers having at least 1000 subscribers. Moreover, the *NOT-TAKE* field specifies that no feedback about customer preferences has to be taken from music sellers.

B. Management of the Service Provider Descriptors

The descriptor of a service provider is filled in by its administrator at registration time. In order to facilitate this activity, we have developed a Trust Management System that offers a graphical user interface for the introduction of the features of the service under specification and the conditions of the trust relationships. This system stores information about all the registered service providers and manages the network of trust relationships by summarizing them, in order to support an efficient propagation of information between applications.

Figure 4 shows a portion of the user interface of the Trust Management System, concerning the introduction of information about an individual service provider. At the right side, the screenshot shows a portion of the registration form (”Nome” - name; ”Subscribers”, ”Anni in internet” - years of activity in internet, etc.). At the left side, a window shows the list of the registered service providers. Figure 5 shows another page, supporting the definition of trust relationships. The service provider administrator is guided in the definition of trust conditions that specify which applications can use the customer information provided by the service under specification.[1] In particular, the system enables the administrator to include/exclude specific categories of applications, require a minimum/maximum number of customers, or number of years of activity in internet, and include/exclude specific marketing areas. Similar pages are generated to support the definition of conditions on the retrieval of customer information from other service providers. The system assists the administrator in the specification of trust relationships by performing consistency checks on the defined trust conditions. For instance, the same condition cannot be specified both in the *GIVE* and the *NOT-GIVE* fields.

Given the trust relationships specified by the administrator (*GIVE*, *NOT-GIVE*, *TAKE* and *NOT-TAKE* fields of the descriptor), the Trust Management System generates three trust relationship lists, *GIVE-IND*, *NOT-GIVE-IND* and *TAKE-IND*, by analyzing the descriptors of the other service providers (e.g., see Figure 6). Specifically:

- The *GIVE-IND* list is generated by selecting the service providers that satisfy at least one *GIVE* condition, that do not satisfy any *NOT-GIVE* condition and that do not trust any untrusted service provider (i.e., the transitive closure of the *GIVE-IND* relationship does not include any untrusted provider).
- The *NOT-GIVE-IND* is generated by subtracting the ap-

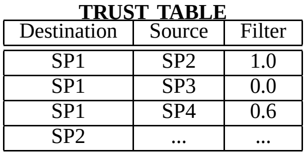
```
ID: SP1;
Trust relationships:
   TAKE-IND: {(SP2, 1), (SP10, 0.5), (SP45, 0), ...}
   GIVE-IND: {SP2, SP10, ...}
   NOT-GIVE-IND: {SP3, SP8, ...}
```

Fig. 6. Trust Relationships between Individual Service Providers.

[1] We assume that the service provider administrator fills in the forms by providing correct data. The provision of false identities is a legal problem that cannot be handled at the technical level.

TABLE I

SUMMARY OF TRUST RELATIONSHIPS FOR INFORMATION SHARING.



TRUST TABLE

Destination	Source	Filter
SP1	SP2	1.0
SP1	SP3	0.0
SP1	SP4	0.6
SP2	...	...

plications in the *GIVE-IND* list from the complete set of registered applications.

- The *TAKE-IND* list includes all the registered service providers and specifies, for each one, the level of trust in the customer information they provide. This is a real number in [0, 1] and has the same meaning adopted in the *TAKE* field of the descriptor. Untrusted service providers have a 0 trust level.

  The level of trust associated to service providers is computed as follows: each service provider that satisfies at least one *TAKE* condition, does not satisfy any *NOT-TAKE* condition and does not trust any service provider satisfying a *NOT-TAKE* condition has level equal to the minimum value associated to the provider by means of the *TAKE* conditions. All the other service providers receive a level of trust equal to 0. For instance, consider the *TAKE* and *NOT-TAKE* conditions reported in the descriptor of *SP1* in Figure 3. A service provider classified both as a *bookSeller* and *movieSeller* would receive a 0 level of trust because it satisfies a condition reported in the *NOT-TAKE* field. Notice that these conditions are evaluated in a pessimistic way (minimum value) because they are associated to the quality and usefulness of the customer information that is going to be received by a service provider. If some characteristics of an application have the potential to introduce noisy data, or irrelevant data, the quality of its contribution is reduced.

The generation of these lists is aimed at presenting details about the trusted and untrusted applications registered for customer information sharing. By exploiting the Trust Management System, the administrator of a service provider SP may inspect and modify (also by overriding the trust relationships that have been defined) the lists of service providers receiving information from SP or providing information to SP. Therefore, the administrator may periodically check the set of registered service providers and update the lists to include and/or exclude new applications. This is important for two reasons: first, the administrator needs to treat individual service providers in a special way (e.g., to trust a provider belonging to a generally untrusted category and vice versa). Second, as time passes, the set of registered applications may change: other service providers may modify their descriptors (a book seller might start to sell music, as well) and new service providers may register.

C. Summarizing Trust Relationships

Although the *GIVE-IND, NOT-GIVE-IND* and *TAKE-IND* lists provide complete information about the trust relationships between pairs of service providers, they fail to support the efficient propagation of the user models at run-time. In fact, each time the User Modeling Agent has to propagate the customer information from a service provider SP_j to another one SP_i, the agent should check:

- whether SP_i satisfies the *GIVE* restrictions specified by SP_j, and
- to which extent SP_i is trusting the information provided by SP_j (trust level in SP_i's *TAKE-IND* restrictions).

In order to support the efficient propagation of information between service providers, we have decided to pre-compile the trust relationships: in the *service provider DB* a *TRUST* table summarizes the trust relationships existing between all the registered service providers; see Table I. The table abstracts from the details of the *GIVE* and *TAKE* relationships, which represent unilateral viewpoints on the propagation of information, and describes the weight of the information provided by the various applications in the generation of the user model for each service provider. More specifically, in the table:

- The *Destination* column represents the service provider receiving the information.
- The *Source* column denotes the service provider that should provide the information.
- The *Filter* column includes real values in [0, 1] and specifies to which extent the information provided by the source application must be taken into account when integrating the customer's preferences to be sent to the destination application. As usual, if the filter takes a value close to 1, this means that the information provided by the source has to be propagated to the destination. Moreover, if the filter is 0, no information has to be propagated.[2]

The *TRUST* table is generated and revised off-line by our Trust Management System. The revision process is launched periodically, in order to update the table according to the changes in the pool of registered service providers; e.g., new registrations, removals, changes in the descriptors.

D. Run-time Customer Information Sharing

The idea behind customer information sharing is that, when an application SP_i invokes the User Modeling Agent to retrieve information about a customer C's preferences, the Agent exploits the *TRUST* table to select the service providers to be contacted. Only the applications whose filter is positive are considered in the generation of the user model and the value of the filter is exploited to merge the preference estimates provided by the applications. Specifically, the User Modeling Agent should retrieve C's preferences from the other registered applications according to the following principles:

[2] The filter takes the 0 value if the destination application is in the *NOT-GIVE-IND* list of the source or if the level of trust between destination and source in the *TAKE-IND* list is 0. Otherwise, the filter takes the trust level specified in the *TAKE-IND* list and thus corresponds to how strongly the destination application trusts the quality of information provided by the source.

1) The bidirectional trust relationships between SP_i and the other applications stored in the *TRUST* table guide the identification of the subset of applications to be considered by the User Modeling Agent and specify SP_i's trust in the provided information (*Filter* field of the table).
2) Within the set of selected applications, only those having C as a registered customer have to be considered.
3) The fact that C has registered in an application SP_j does not mean that SP_j has already acquired any preference information about C.

In order to take the first two factors into account, the User Modeling Agent consults the *TRUST* table to select a set of candidate applications and it queries the *user model DB* to identify the applications that have C as a registered customer. The agent exploits the *Filter* information stored in the *TRUST* table to tune the influence of their customer information in the generation of the user model. The trust level has to be taken into account when combining the contribution of the applications to the generation of the model. Ideally, the trusted applications should stronfly influence the generation of the user model, while the less trusted ones should marginally influence the process.

As far as the third factor is concerned, the contribution to the generation of the user model carried by each application A must be also tuned according to the confidence of A in the provided information (confidence degree assigned by the application, given the amount of evidence about the customer at disposal). As specified in the *customer preference ontology*, each customer preference has an associated confidence degree, describing the reliability of the information, i.e., whether there is evidence about the provided information or not.

We have selected a weighted addition formula to combine the information about the customer preferences provided by the applications invoked by the User Modeling Agent. For each requested preference P, the agent combines the interest estimates provided by the trusted applications SP_j as follows:

$$Int\_P = [\sum_{j=1}^{n} MIN(trust_{ij}, conf_j) * Int\_P_{SP_j}]/\delta \qquad \textbf{(i)}$$

where $\delta = \sum_{j=1}^{n} MIN(trust_{ij}, conf_j)$

In the formula:

- Int_P is the interest value for P generated by the User Modeling Agent, given the contribution of the invoked service providers;
- n is the number of invoked applications;
- $trust_{ij}$ represents how strongly SP_i trusts SP_j (i.e., it is the *Filter* associated to SP_j in SP_i's *TRUST* table);
- $conf_j$ denotes the confidence associated to the interest by SP_j;
- δ is applied to normalize Int_P in [0, 1].

For each invoked application SP_j, the contribution to the computation of the interest value for P is thus weighted according to SP_i's trust level in SP_j and to SP_j's confidence in the estimated preference. The minimum of the two values is exploited to define the impact of the estimate according to a *Fuzzy AND*.

The formula (i) enables the User Modeling Agent to merge the information provided by the various applications according to the service providers' requirements, but also on the basis of a subjective evaluation of the reliability of the provided information. As confidence values are associated to individual preferences, they may change from invocation to invocation, depending on the observations of the customer behavior carried out by the applications.

V. Discussion and Related Work

Some policy-based approaches [16] have been proposed to manage the trust relationships between applications and to regulate the access to shared resources and data. For instance, the framework described by Kagal and colleagues [17] supports the automatic and distributed management of access rights to resources and information. The framework is implemented in a language supporting the specification of deontic concepts, such as rights and prohibitions to perform actions. Suitable *security agents* apply the defined policies to grant or cancel access and delegation rights to groups of agents in a controlled way, by delivering certificates.

Indeed, the purpose of our work differs from Kagal et al.'s work [17], [18], in relation to the type of rights we aim at regulating.

- Kagal et al. control different types of actions that the applications may perform on the resources, such as "reading", "writing", "executing" a file. Instead, we are only concerned with "reading" rights.
- At the same time, however, our framework enables the applications to define restrictions on the type of information they want to receive and controls the information flow accordingly.

VI. Conclusions and Future Work

We have presented a framework for customer information sharing that supports the controlled propagation of information among service providers. Our framework includes a registration service (the Trust Management System) exploited by service providers in order to join the pool of applications sharing information with one another. Moreover, the framework includes a User Modeling Agent that controls the information flow between applications and reconciles the information provided by the various service providers in order to generate the preference information needed by the requesting application.

We have developed a proof-of-concept implementation of the customer information sharing framework that supports the service provider administrator in the introduction of information about service providers and trust relationships. The framework is based on Java and uses JDBC technology to connect to the database where the trust information is stored in the corresponding tables.

Our framework handles bidirectional trust relationships to address the fact that service providers may want to:

- control the dissemination of information by imposing restrictions on the service providers that will receive data;

- impose restrictions on the service providers from which they want to receive information, in order to filter out irrelevant information sources available through the information sharing service.

As already specified, we have left the management of the customers' privacy preference aside, assuming that the customers do not impose restrictions on the dissemination of their personal data. In our future work, we will extend our proposal to the treatment of customer preferences, which can be done without major architectural changes. Specifically, taking the P3P specifications into account, the *user model DB* handled by the User Modeling Agent could be extended to store the individual customer's privacy preferences. Moreover, the overall service should require that, at registration time, the service providers publish their own P3P privacy policies. Having this information available, the User Modeling Agent could propagate the customer information between applications by taking into account not only the trust relationships, but also possible constraints imposed by the individual customer.

In our future work we will analyze the ontology issues concerning the binding between the service providers' local representations of the customer information and the one adopted in the customer information sharing service. Our goal is the development of an ontology binding tool supporting the administrator of a service provider to define the correspondences between the customer preferences defined in the application and those exploited by the main user model for information sharing.

In our future work we will also study the possibility of distributing the information sharing service for efficiency and reliability purposes. For instance, an interesting solution to study is a distributed User Modeling Agent in the line of peer-to-peer sharing networks, where the applications directly contact other trusted applications to gather customer profile information.

References

[1] P. Resnick and H. Varian, Eds., *Special Issue on Recommender Systems*. Communications of the ACM, 1997, vol. 40, no. 3.

[2] J. Fink and A. Kobsa, "A review and analysis of commercial user modeling servers for personalization on the World Wide Web," *User Modeling and User-Adapted Interaction, Special Issue on Deployed User Modeling*, vol. 10, no. 2-3, pp. 209–249, 2000.

[3] M. Maybury and P. Brusilovsky, Eds., *The adaptive Web*. Communications of the ACM, 2002, vol. 45, no. 5.

[4] R. Burke, "Hybrid recommender systems: survey and experiments," *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 289–322, 2002.

[5] M. O'Connor, D. Cosley, J. Konstan, and J. Riedl, "PolyLens: a recommender system for groups of users," in *Proc. European Conference on Computer Supported Cooperative Work (ECSCW 2001)*, Bonn, Germany, 2001.

[6] D. Billsus and M. Pazzani, "A personal news agent that talks, learns and explains," in *Proc. 3rd Int. Conf. on Autonomous Agents (Agents '99)*, Seattle, WA, 1999, pp. 268–275.

[7] A. Kobsa, "Personalized hypermedia and international privacy," *Communication of the ACM*, vol. 45, no. 5, pp. 64–67, 2002.

[8] A. Kobsa and J. Schreck, "Privacy through pseudonymity in user-adaptive systems," *ACM Transactions on Internet Technology*, vol. 3, no. 2, pp. 149–183, 2002.

[9] W3C, "Platform for Privacy Preferences (P3P) Project," http://www.w3.org/P3P/.

[10] "RosettaNet ebusiness standards for the Global Supply Chain," http://www.rosettanet.org/RosettaNet/Rooms/DisplayPages/LayoutInitial.

[11] UbisWorld, "Ubiquitous User, Modeling for Situated Interaction," http://www.u2m.org/.

[12] D. Heckmann, "A specialised representation for ubiquitous computing," in *Proc. Workshop on User Modelling for Ubiquitous Computing*, Johnstown, PA, 2003, pp. 26–28.

[13] L. Ardissono, C. Gena, P. Torasso, F. Bellifemine, A. Chiarotto, A. Difino, and B. Negro, "Personalized recommendation of TV programs," in *LNAI 2829. AI*IA 2003: Advances in Artificial Intelligence*. Berlin: Springer Verlag, 2003, pp. 474–486.

[14] L. Ardissono, C. Gena, P. Torasso, F. Bellifemine, A. Difino, and B. Negro, "User modeling and recommendation techniques for personalized Electronic Program Guides," in *Personalized Digital Television. Targeting Programs to Individual Users*. Kluwer Academic Publishers, 2004.

[15] Liberty Alliance Developer Forum, "Liberty alliance project specifications," http://www.projectliberty.org/specs/, 2004.

[16] M. Sloman, "Policy driven management for distributed systems," *Journal of Network and Systems Management*, vol. 2, no. 4, pp. 333–360, 1994.

[17] L. Kagal, S. Cost, T. Finin, and Y. Peng, "A policy language for pervasive systems," in *Proc. 4th IEEE Int. Workshop on Policies for Distributed Systems and Networks*, Lake of Como, Italy, 2003.

[18] L. Kagal, T. Finin, and A. Joshi, "A policy based approach to security for the Semantic Web," in *Proc. 2nd Int Semantic Web Conference (ISWC2003)*, Sanibel Island, FL, 2003.