

Customer Information Sharing between E-commerce Applications

Liliana Ardissono, Marco Botta, Luca Di Costa,
Giovanna Petrone

*Dipartimento di Informatica
Università di Torino*

Fabio Bellifemine, Angelo Difino and
Barbara Negro

Telecom Italia Lab

Personalization in e-commerce

Product recommendation enhanced by
means of personalization techniques

→ CF, CBF to recommend goods

→ Bootstrapping issue: what about
new customers?

Plan of the talk

1. Open issues in customer information sharing between applications
2. Architecture of customer information sharing framework
3. Trust Management System
4. Controlled propagation of customer information between service providers
5. Conclusions and future work

Sharing customer information

Preference acquisition speeded up if service
providers share customer information

→ New customers may be treated as known
ones

→ Customer information is precious

→ User data (features and inferences)

→ Clickstream data

Trust between providers

Trust important for customer
information sharing

→ Service provider wants to

→ give data to trusted parties

→ take data from trusted sources (noise,
low quality data, ...)

Goal of our work

Information sharing framework supporting

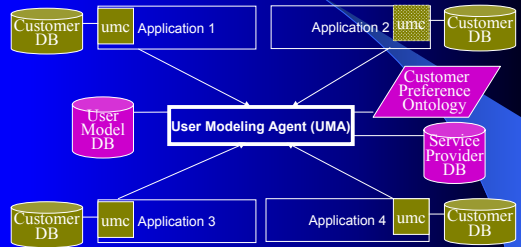
→ Specification of mutual trust
relationships between service providers

→ Propagation of customer information
between service providers, respecting
trust relationships

Open issues

- Taking customer privacy preferences into account (see P3P)
- Ontology mapping between service providers (see P3P and Semantic Web – UbisWorld: Ubiquitous computing for User Modeling)
- **Trust relationships between service providers (information propagation)**

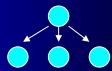
Architecture of framework



Customer Preference Ontology

Representation of preferences adopted in UMA and in registered SPs

- Preference: $\langle \text{Interest degree (in } [0, 1]), \text{ Confidence degree (in } [0,1]) \rangle$
- Books
 - history: (Int: [0, 1], Conf: [0,1])
 - science: (Int: [0, 1], Conf: [0,1])
- Music
 - rock: (Int: [0, 1], Conf: [0,1])
 - disco: (Int: [0, 1], Conf: [0,1])



Service Provider DB

Stores information about registered service providers (SP descriptors)

- SP data (name, ..., categorization)
- Trust relationships with other SPs (in input and in output)

SP Descriptor

- **Identification data** (ID, name, URL, ...)
- **Categorization**: one or more SP categories (bookSeller, movieSeller, ...)
- **Features** (numberOfSubscribers, ...)
- **Trust relationships**
 - TAKE conditions
 - NOT-TAKE conditions
 - GIVE conditions
 - NOT-GIVE conditions

Trust Relationships - TAKE

conditions on applications from which SP wants to **receive** info, + degree of trust in info

- TAKE = {(bookSeller OR movieSeller) AND nrOfSubscribers > 1000, 1), ...}

Trust Relationships - GIVE

conditions on applications to which SP wants to disseminate info

→ GIVE = {(bookSeller OR movieSeller), ...}

Trust Relationships NOT-TAKE, NOT-GIVE

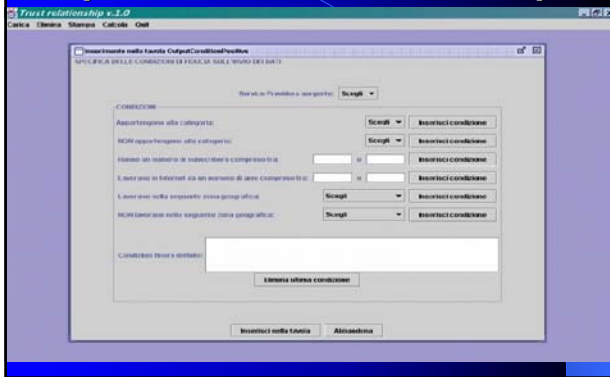
NOT-TAKE: conditions on applications from which SP does NOT want to receive info

→ NOT-TAKE = {musicSeller, ...}

NOT-GIVE: conditions on applications to which SP does NOT want to give info

→ NOT-GIVE = {insuranceAgent, ...}

Specification of relationships



Explicit Trust Management

Trust policies based on SP properties not transparent or flexible enough

→ Explicit management of SP lists generated on the basis of trust policies

SP administrator can check which SPs satisfy trust policies and decide whether they are really trusted parties!

Trust Management: GIVE-IND

List of SPs which may receive info. All SPs which

1. satisfy at least one GIVE condition
2. do not satisfy any NOT-GIVE condition
3. do not trust any untrusted SP



Trust Management: TAKE-IND

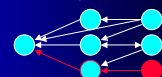
Specifies level of trust ([0, 1]) in customer info provided by SPs

if SP

1. satisfy at least one TAKE condition
2. does not satisfy any NOT-TAKE condition
3. does not take info from SPs satisfying a NOT-TAKE condition

then level = min level attributed to SP by TAKE conditions

otherwise level = 0



Trust Management System - I

Tool for service administrator

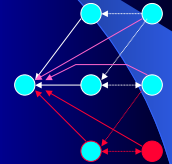
- generates lists (revised periodically)
- displays lists to administrator
- enables administrator to
 - validate content of lists (before customer data goes to new list members)
 - modify the lists to manually include/exclude SPs

Trust Management System - II

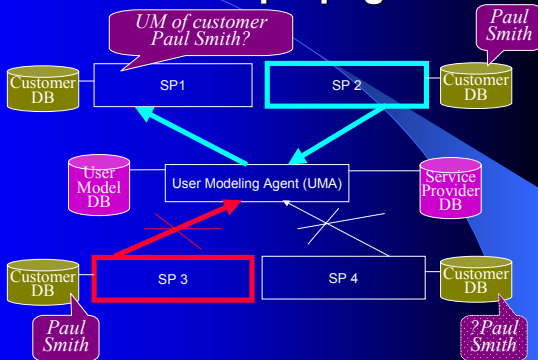
- Summarizes trust relationships between all SPs for efficiency purposes

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

TRUST TABLE



Customer info propagation - I



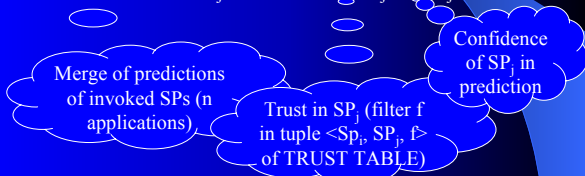
Customer info propagation - II

UMA

- Selects from TRUST TABLE SPs with positive filter, who know customer
- Retrieves UM from SPs and tunes contribution of each SP according to
 - filter in TRUST TABLE
 - confidence of SP in prediction
- Trusted applications have stronger impact, if confident in their predictions

Customer info propagation - III

$$Int_P = \frac{\sum_{j=1..n} MIN(trust_{ij}, conf_j) * Int_P_{SP_j}}{\sum_{j=1..n} MIN(trust_{ij}, conf_j)}$$



Conclusions

Trust Management System

- Supports service administrator in specification of trusted parties
- Enables explicit control of trusted party lists

User Modeling Agent

- Enforces propagation of customer info according to mutual trust relationships between SPs

Overall framework supports SPs in secure information sharing

Future work

- Extend framework to take **customer privacy preferences** into account (additional checks in UMA when propagating information between SPs)
- Study possibility to **distribute UMA** for efficiency purposes (direct interaction between trusted applications)

User Model DB

- Stores identification data of customers
- For each customer registration (to get services of SP)
- If no global identifier (passport), customer treated as different people