A Multi-Tiered Recommender System Architecture for Supporting E-Commerce

L. Palopoli and D. Rosaci and G.M.L. Sarnè
Motivation

- *Recommender Systems* are tools able to explore the Web space for promoting e-Commerce activities by supporting customers with recommendations.

- *Recommender Systems* can provide suggestions for users’ purchases based on a representation of their interests and preferences.
Motivation

- Many Recommender Systems (RSs) are centralized, lack in efficiency, scalability and customers' privacy (due to the centralization of personal information)
- Other RSs are distributed and require a computational overhead excessive for many devices (e.g. mobile devices)
- The most part of RSs assume homogeneous system components making difficult for users to add personal knowledge in the system
This work presents a distributed agent-based RS, called DAREC (Distributed Agent Recommender for E-Commerce).

DAREC is able to generate very effective suggestions without a too onerous computational task.

DAREC introduces significant advantages in terms of openness, privacy and security in all a B2C process.
Different behavioral models describe the phases of a B2C process, as the well known Consumer Buying Behavior (CBB) model based on six stages:

- **Need Identification**: A user identifies his/her needs
- **Product Brokering**: A user searches for products that satisfy his/her identified needs
- **Merchant Brokering**: When the consumer decides what to purchase, he/she tries to identify a suitable merchant selling the chosen goods or services
- **Negotiation**: Transaction terms are fixed
- **Purchase and Delivery**: The customer finalizes the purchase choosing a payment option and a delivery modality
- **Service Evaluation**: The customer evaluates his/her satisfaction level about his/her purchase
DAREC

- In DAREC each customer is assisted by 3 specialized software agents, each of which, autonomously of the other agents, deals with a different CBB stage

- Each agent runs on a different thread on the customer's client in order to improve the efficiency of the overall process
When a customer needs to interact with a customer for Need Identification purposes, his $NI(PB,MB)$-agent simply interacts with the other 's $NI(PB,MB)$-agent.

When there is a unique customer's agent, it can execute only one activity at time.

In DAREC the other agents of and are free for other activities and in this way:

- DAREC can increase the distribution degree of the RS
- DAREC can generate effective recommendations without a too onerous computational task
- DAREC introduces significant advantages in openness and privacy
DAREC

- Each customer's agent can interact with DAREC sellers' sites, each one assisted by a seller agent provided with:
  - a product catalogue;
  - the customers' profiles encoding the preferences of each customer that visited the site in the past

- The customer’s agent interaction with the seller agents interaction of permit to generate:
  - content-based (CB) recommendations for the customer;
  - personalized site presentations of the products for the site visitors

- The interaction with the other agents permit to generate collaborative filtering (CF) recommendations
The DAREC community shares a common dictionary storing the names of basic product categories of interest and their reciprocal relationships.

Each agent encodes in a profile encodes all the information necessary to perform its task.

In order to promote collaboration between agents the information stored in a yellow page data structure are used.
The Category Dictionary

- A *Category Dictionary* $D$ is represented as a direct labeled graph $(G, A)$, where:
  - for each category there is a node called associated with a label denoted by
  - for each there is a link $< , , t>$ oriented from to and labeled by $t$, where $t$ is the type of the link that can be:
    - *isa-link*, denoted $< , , ISA>$, iff all the products belonging to also belong to
    - *synonymy-link*, denoted $< , , SYN>$, iff both all the products belonging to also belong to and vice versa
    - *overlap-link*, denoted $< , , OVE>$, iff there exist some product of that also belong to , and vice versa. Note that if two categories are synonymy-linked, they are also overlap-linked
    - *commercial-link*, denoted $< , , COM>$, iff we suppose that the customers usually purchase both products belonging to and
Personal Profiles

- In each CBB stage, a customer is assisted by an agent storing in a profile all the customer's information to handle that CBB stage.

- Note that a customer can perform a CBB stage without performing next stages. Thus the categories in the profiles are subsets of those in

- Finally, each category belongs to either the common dictionary or to a personal customer's category (understandable to the other agents being in a general relationship with at least another category belonging to $D$).
Each seller $S$ is associated with a seller agent $s$ that stores in its site profile, for each category, all products belonging to that category that are offered by the seller and for each product stores some commercial information and the list of the past customers interested in it in the past.

Is a set of category dictionaries, one for each customer $c$ implemented as a sub-graph of $c$'s NI-profile, containing those categories such that $c$ desires to make public.
Agents’ Behavior

- A newcomer should build an initial profile by adding the categories with an initial interest degree, the visibility mode and their relationships.

- Moreover he/she could add some personal category with name, path in and linked with at least a

- The customer $c$ for each product can:
  - (A1) watch the product
  - (A2) select the product for examining the seller’s offer
  - (A3) purchase the product
Agents’ Behavior

- Each action performed by c implies a call to the agents NI, PB and MB that automatically update their profiles and:
  - The NI-agent is called for the category
  - If it is added therein with an initial interest value
  - Then and its interest value are added to and
  - Otherwise
    - if its interest value is updated to \((\text{with a=A1, A2, A3})\) it is arbitrarily set by c to weight the performed action
  - The value is then passed to the agents PB and MB for updating their profiles
Agents’ Behavior

- The client calls the PB-agent to pass the product $p$
  - If then it is added to the list with an initial interest value and their insertion in the list is required
  Otherwise
    - If its interest value is updated to and passed to $MB$ for updating its profile

- The client calls the MB agent, passing the seller $s$.
  - If it is added to the list with and an initial score
  Otherwise
    - increased by 1 and the score is updated to
Agents’ Behavior

- Periodically:
  - The value associated with the NI (PB, MB)-profile, after a time period passed from its last update, is decreased of, a c's parameter ranging in [0,1].
  - The seller agent updates its list after each customer's action that involves a product.
  - If a new element is added to the list, and the number of transactions is increased.
  - If then is updated by the agent s increasing and inserting p if it is absent.
The Recommender

- When c selects the tab *Recommender* in his client, then some suggestions are generated for him and visualized in a page having a section for each supported stage.

- Suggestions are generated by the each agent in a *cascade* mode.

- In order, c chooses:
  - A category from those in section ``Recommended Categories``
  - A set of products is suggested in section ``Recommended Products``
  - A set of merchants selling that product is suggested in section ``Recommended Merchants``
An example
The Recommender

- The NI-agent suggests to c a set of categories visualized in the client section *Recommended Categories* in the 3 list-boxes:
  - **Visited Categories** contains categories selected with a CB approach from the NI-profile based on the c's activity
  - **Unvisited Categories** that are unknown to c, but considered interesting by his NI-agent interacting with each site agent that he visited in the past, by means of a *relationship-based* mechanism
  - **Suggested by Similar Customers**, with a CF technique, on the whole EC customer's navigation history by the c's NI-agent collaborating with the NI-agents of customers similar to c for interests.

The c's NI-agent computes the Jaccard similarity degree between the set of nodes stored in its profile and each public repository (storing, for each DAREC customer, his public interest profile) to consider those categories unknown to c.
The Recommender

- The PB-agent (MB-agent) suggests, in the section *product recommendations* (*merchant recommendations*) a set of *products* (*sellers*) belonging to a category *cat* selected by *c* from the recommended *categories* (*products*) on his client:
  - **Visited Products (Merchants)** contains *products* (*merchants*) of the category (*...*), ordered by score
  - **Unvisited Products (Merchants)** obtained by exploiting a collaboration between the *c*’s PB (resp. MB)-agent and the seller agent of each site that *c* visited in the past to select (*...* in their profiles that are unvisited by *c*
  - **Suggested by Similar Customers**, based on the collaboration between the PB (MB)-agents of *c* and of other similar customers. Each PB (MB)-agent of these customers sends to *c*’s PB (MB)-agent its set of products (merchants) that is added to this list
The Recommender

- The PB (resp. MB)-agent shows the products (merchants) belonging to the listbox *Visited Products* (*Visited Merchants*), ordered by value, and the products (merchants) belonging to the listboxes *Unvisited Products* (*Unvisited Merchants*) and *Suggested by Similar Customers*, ordered alphabetically.

- Each seller agent exploits its profile to personalize the site presentation for each customer $c$ that is visiting it based on the he already visited in the past.

- Using such information, personalizes its home page for $c$ by visualizing all the ordered by interest value, increments the value for the current $p$ and $c$ to consider his current visit. Otherwise, for the first time $c$'s visit the default home page is visualized.
Efficiency of DAREC

- In terms of efficiency, for a community of $n$ customers and $m$ sellers, a unique centralized agent managing all the three phases has computational cost of $\ldots$, where:
  - $C$ is the number of contemporary sessions running for a CBB stage
  - $O$ is the number of operations needed for a user to manage a CBB stage
- In DAREC, for a given CBB stage, each user's agent can deal with more different issues running on different threads.
- Let $T$ be the multi-threading degrees for a specific CBB stage and let $H$ be the computational overhead due to the communications between the local agents. In this way the computational cost for a CPU will be
Efficiency of DAREC

• The computational advantage (\( \mathcal{A} \)) of DAREC, is equal to

\[
\mathcal{A} = \frac{\text{operations}}{\text{parallel threads}}
\]

where if, for simplicity, we assume that \( \text{operations} \), \( \text{parallel threads} \), the above formula becomes

\[
\mathcal{A} = \frac{\text{operations}}{\text{parallel threads}}
\]

• Therefore, the advantage of using DAREC is perceivable with a small multi-threading contribution (i.e. high values of \( \mathcal{A} \)) in presence of a reasonably high number of operations (i.e. an intense EC activity), while for a high multi-threading activity the advantage shows up also with a small number of operations (N)
Effectiveness of DAREC

- The time exploited to perform B2C processes in serial and multi-threading way has been compared by means of a software appositively designed.

- We considered a period of 2 hours where a set of 500 customers finalize all their B2C processes with a purchase dealing with a merchant population ($M$) of 10 units.

- The merchant has to satisfy also the requests due to other customers that could absorb significant servers resources. It is taken into by means of an overhead ($O$) of requests for second, randomly shared among the merchants.

- Finally, a lot of different of communication, computational and behavioral parameters have been tuned to model realistic B2C processes.
Effectiveness of DAREC

- The same values for the parameters have been used in order to compute the average time (seconds) needed to perform a purchase process in a:
  - multi-threading (modality 
  - serial (modality 

$NP$ is the number of purchases, randomly fixed, performed in the considered test session
Effectiveness of DAREC

- The average serial ( ) and multi-threading ( ) times (in sec.) needed to carry out a B2C process depending on the Overhead by considering 500 Customers and 10 Merchants

<table>
<thead>
<tr>
<th>Overhead</th>
<th>0</th>
<th>5</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>90</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Gain</td>
<td>24.61</td>
<td>36.12</td>
<td>44.63</td>
<td>56.43</td>
<td>63.98</td>
<td>69.19</td>
<td>73.05</td>
<td>76.39</td>
<td>78.39</td>
<td>80.75</td>
<td>82.88</td>
<td>83.61</td>
</tr>
</tbody>
</table>
The experimental results shown in Figure confirm that the DAREC approach consumes in average about the 25% of time less than the serial approach in performing a purchase in absence of overhead and when the overhead grows also grows with it while is almost uniform.

This behavior is due to the fact that changes in the number of merchants, overheads and so on, have a minimal impact on and very high impact on

This because, in average, each merchant's server is busy to satisfy the customers' requests and grows with the level of saturation of the merchants' servers worsening the quality of their service.
Conclusions

- In this paper, we have presented the DAREC architecture that introduces novel, original characteristics with respect to other recommender systems.

- DAREC allows to the different CBB stages of an EC process to be assigned to a different agent creating a tier of specialized agents.

- This architecture reduces the computational cost for the device on which the local agents run and the specialized agents improve the users' knowledge representations, the openness of the system and the privacy degree.
A Multi-Tiered Recommender System Architecture for Supporting E-Commerce

L. Palopoli and D. Rosaci and G.M.L. Sarnè

THANKS