From Collaborative Filtering to Implicit Culture: a general agent-based framework

Enrico Blanzieri
ITC-IRST
Via Sommarive 18
Povo, 38050 Trento - Italy
blanzier@itc.it

Paolo Giorgini
DISA - University of Trento
Via Inama 5
38100 Trento - Italy
pgiorgini@cs.unitn.it

ABSTRACT

Collaborative Filtering bases its effectiveness as a recommender system on ratings about a set of items provided by a set of users. In our perspective, an agent behaves as a member of a group would do (the agent implicitly belongs to the same "culture" of the group) without extra-effort or direct interaction. In this paper, we introduce the concept of Implicit Culture and propose a general architecture for Systems for Implicit Culture Support. We show how Collaborative Filtering can be considered as an instance of our architecture, and finally, we consider the related work.

1. INTRODUCTION

Given the problem of information overload, the building of recommender systems is a mayor issue. Collaborative Filtering (see [4] for a recent reference) demonstrated to be an effective approach from an applicative point of view. However, the ideas underlying Collaborative Filtering have a greater scope than filtering itself. In this paper we capture those ideas in the notion of Implicit Culture and we propose an agent-based framework for it.

When an agent begins to act in an environment without enough knowledge or skills, its behavior will be far from optimal. The problems that the new agent has to face are even more complex if some other agents are active in the same environment. They would probably have more knowledge and would be more skilled. Moreover, they might not be willing to share their knowledge and sometimes not even able to represent or communicate it.

It is easy to find several examples of this problem. An oversimplified version of it occurs when a new user logs on a system and she does not know the name of the nearest printer. Another example can be observed when browsing the web on a not-familiar topic, it is hard to locate what the other users considerate the relevant resources. Again, the problem arises when a buyer or a seller starts to operate on an unknown market wondering about reasonable prices and reliable partners. In all these examples the agent lacks the knowledge that the other agents - namely users, web-surfers or operators - have about their environment and about each other.

In order to improve its behavior, the new agent should act consistently with the culture of the group. In fact, in this "new kid in town" scenario the agent is enable to cope with the environment and with the other agents. More depressingly, a group of agents have the knowledge and actively exploit it. In the case of humans the phenomenon is sometimes referred as "cultural shock". In fact, knowledge about the environment and about the behaviors of the agents is part of their culture and that is what the new agent lacks.

The problem of having the new agent acting consistently with the knowledge and behaviors of the group could be solved by improving the capabilities of the agent in terms of communication, knowledge and learning. The first solution is to "just ask someone" and , in a agent setting, it is *not* a simple solution. It is necessary to know what to ask (knowledge about the problem), how to ask (a language for expressing the problem), and who to ask to (some brokering facility). More fundamentally, it is also necessary to know that one has a problem in the first place, and to have its solution among the goals. The second possible solution is to represent the relevant knowledge and provide it to the agent. If the knowledge required is objective and relatively static, the representation can be done observing the environment and describing it. Building ontologies is a common way of addressing this problem. Unfortunately, the environment can be partially unknown and intrinsically dynamic. As a third option, it is possible to equip the agent with both observational and learning capabilities and acquire skills by imitation of the other agents. As a drawback, these capabilities are rather complex and their application requires resources.

When the environment is partially under control, the problem can be tackled in a very different way. Instead of working on the agent capabilities, it is possible to modify the view that the agent has of the environment and consequently its actions. In fact, changing in a proper way the set of possible actions that the agent can do in the environment can lead the agent to act consistently with the behavior a member of the group would have. The group itself can have optimized its behavior on the particular environment. Moreover, neither the new agent nor a member of the group is required to know about it and so they share the same culture in an implicit way.

In the present paper we introduce the concept of Implicit Culture for describing the situation in which agents behave according to a cultural schema or contribute to produce the cultural schema without the need to know about the group, its members or their behavior. Moreover, we propose an architecture for systems aimed to support the emergence of an Implicit Culture on a group of agents We show how Implicit Culture solves the problem of persistence of the requirements of a system of agents without affecting their level of autonomy, reducing the undesired behaviors and exploiting the useful ones. The architecture is general and covers Collaborative Filtering [4] as a particular case.

The paper is organized as follows: the next section 2 introduces the concept of Implicit Culture; section 3 presents an architecture for supporting it; section 4 shows some instances of the architecture; and finally, sections 5 and 6 describe related works and draw conclusions respectively.

2. IMPLICIT CULTURE

A group of agents effectively acting into an environment exploit a great amount of knowledge and skills. When new agents are introduced in the environment they face the problem of acquiring the necessary knowledge. The problem of the new agents would be solved if they acted in a way consistent with the knowledge and behaviors of the group, namely its culture. If the environment is under control and modifiable it is possible to obtain the same effect without the need for the agents to know about the group and its behavior. We call this phenomenon Implicit Culture.

We assume that each agent is acting in an environment composed of objects and other agents. Actions have as arguments objects, as in *offer(book1, price1)* or *demand(book2, price2)*, agents, as in *look_for(buyer)* or *ask_about(seller)* or both objects and agents, as in *send(message, seller)*.

Before executing an action an agent faces a scene formed by a portion of the environment, namely objects and agents, and actions that are possible in it. For example, an agent *buyer* faces *seller1, seller2, book1, gadget1, price1, price2* and can perform *buy_from(seller1,book1,price1)*, *buy_from(seller2,gadget1,price2)* and *buy_nothing()*. Hence, an agent executes an action in a given situation, namely the agent facing a scene at a given time, so the agent executes situated actions. The agent *buyer* executed the action *buy_from(seller1, book1,price1)* while he was facing the scene composed of *seller2*, *price2* and the possible actions.

After a situated action has been executed the agents face a new scene. At a given time the composition of the new scene depends on the environment and on the situated executed action. If *buyer1* performs *buy_from(seller1,antique_book1, price1)*, and *buyer2* performs *do_nothing()*, both *buyer1* and *buyer2* will have the scenes they face changed for *antique_book1* is not on sale anymore. If *seller* performs *sell_to(buyer1,antique_book1,price1)* the next scene it faces will not include *antique_book1*.

The situated executed action that an agent chooses depends on its internal and unaccessible states and in general it is not deteministically predictable. Rather, we assume it can be characterized in terms of probability and expectations. As an example, given a *buyer* facing a scene in which it can perform *buy_from(seller1,book1,unreasonable_price)*, *buy_from(seller2, book1, low_price)* or *buy_nothing()* the expected situated action can be *buy_from(seller2,book1,low_price)*.

Given a group of agents let us suppose that there exists a theory about their expected situated actions. If the theory is consistent with the executed actions of the group, it can be considered a validated cultural constraint for the group. The theory captures the knowledge and skills of the members about the environment. For instance:

$$\begin{aligned}\forall x, y \in Group, book \in Books :&\\ &execute(x,\ buy\_from(y, book, p)) \wedge\\ &execute(y,\ sell\_to(x, book, p)) \rightarrow\\ &p \in [low(book), high(book)]\end{aligned} \tag{1}$$

expresses that, for all agents of the group and all books, if a buyer buys a book from a seller (and the seller sells the book to the buyer) then the price of the book will be reasonable, i.e. $low(book) \leq p \leq high(book)$. With this theory we could predict that the situated executed action of *buyer* will be the expected executed action *buy_from(seller1,book1,reasonable_price)* given the fact that *reasonable_price* $\in$ [*low(book),high(book)*].

If a set of new agents performs actions that satisfy the validated cultural constraints of the group the problem of their suboptimal behavior with respect to the group is solved. To have a group of agents such that their actions satisfy a validated cultural constraint of another group with no need to know about it, realizes what we call Implicit Culture. The actions of a *seller* and a *buyer* are far more effective if they face only offers and demands at reasonable prices, and that is true even if they do not know the cultural constraint.

A system for Implicit Culture Support has the goal of establishing an Implicit Culture phenomenon. It reaches the goal by building validated cultural constraints from observations of situated executed actions, and presenting scenes to the agents such that their expected situated actions satisfy the cultural constraint.

3. AN ARCHITECTURE FOR IMPLICIT CULTURE SUPPORT

In this section we present a formal definition of Implicit Culture, a general architecture for Systems for Implicit Culture Support and one example.

3.1 Basic definitions: scenes, situations and culture

Let *agent_name*, *object_name* and *action_name* be strings, we define: the *set of agents* $\mathcal{P}$ as a set of *agent_name* strings; the *set of objects* $\mathcal{O}$ as a set of *object_name* strings; and the *environment* $\mathcal{E}$ as a subset of the union of the set of agents and the set of objects, i.e., $\mathcal{E} \subseteq \mathcal{P} \cup \mathcal{O}$.

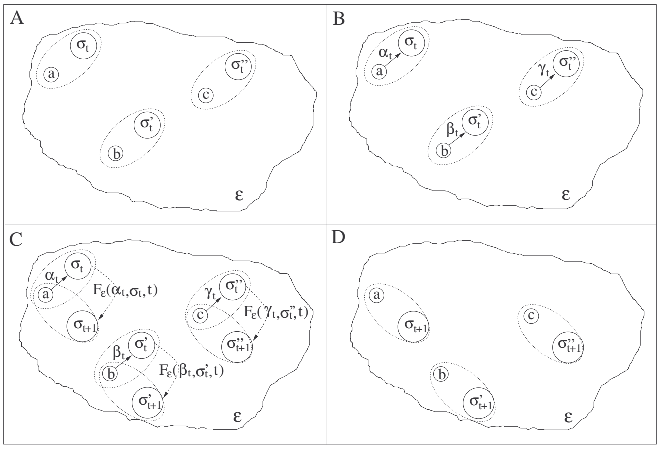

Figure 1: The environment $\mathcal{E}$

Let E a subset of the environment ($E \subseteq \mathcal{E}$) and s an *action_name* string, we define:

- an *action* α as the pair $\langle s, E\rangle$, where E is the *argument* of α ($E = arg(\alpha)$).

Let $\mathcal{A}$ be a set of actions, $A \subseteq \mathcal{A}$ and $E \subseteq \mathcal{E}$, we define:

- a *scene* σ as a pair $\langle E, A\rangle$ where, for any $\alpha \in A$, $arg(\alpha) \subseteq E$. α is said to be *possible in* σ;
- the *scene space* $\mathcal{S}_{\mathcal{E},\mathcal{A}}$, as the set of all scenes.

Let $a \in \mathcal{P}$, α an action and σ a scene:

- a *situation* at the discrete time t is the triple $\langle a, \sigma, t\rangle$. We say that a *faces* the scene σ at time t;
- an *execution* at time t is a triple $\langle a, \alpha, t\rangle$. We say that a *performs* α at time t;
- an action α is a *situated executed action* if there exists a situation $\langle a, \sigma, t\rangle$, where a performs α at the time t and α is possible in σ. We say that a performs α in the scene σ at the time t.

When an agent performs an action in a scene, the environment reacts proposing to the agent a new scene. The relationship between situated executed action and new scene depends on the charateristics of the environemnt, and in particular on the laws that describe its dynamic. We suppose that it is possible to describe such relationship by an environment-dependent function defined as follows:

$$F_{\mathcal{E}} : A \times \mathcal{S}_{\mathcal{E},\mathcal{A}} \times T \rightarrow \mathcal{S}_{\mathcal{E},\mathcal{A}} \qquad (2)$$

Given a situated executed action α_t performed by an agent a in the scene σ_t at the time t, $F_{\mathcal{E}}$ determines the new scene σ_{t+1} ($= F_{\mathcal{E}}(\alpha_t, \sigma_t, t)$) that will be faced at the time $t+1$ by the agent a.

Figure 1 presents how the function $F_{\mathcal{E}}$ works. Particularly, Figure 1.A show the environment $\mathcal{E}$ in which three agents a, b, and c face the scenes σ_t, σ'_t, and σ''_t respectively (the ellipses indicate the three different situations). At the time t the three agents perform respectively the actions α_t, β_t, and γ_t (Figure 1.B). The function $F_{\mathcal{E}}$ changes the scenes so that at the time $t+1$ the agents face the scenes σ_{t+1}, σ'_{t+1}, and σ''_{t+1} (Figure 1.C,D).

While $F_{\mathcal{E}}$ is supposed to be a deterministic function, the action that an agent a performs at time t is a random variable $h_{a,t}$.

Given an agent $a \in \mathcal{P}$ and a situation $\langle a, \sigma, t\rangle$:

- the *expected action* of the agent a is the expected value of the variable $h_{a,t}$, that is $E(h_{a,t})$;
- the *expected situated action* of the agent a is the expected value of the variable $h_{a,t}$ conditioned by the situation $\langle a, \sigma, t\rangle$, that is $E(h_{a,t} | \langle a, \sigma, t\rangle)$.

Given a set of agents $P = \{a_i\} \subseteq \mathcal{P}$, we denote with the vector $\bar{\alpha}_t = \{\alpha_t[i]\}$ the actions they perform at time t respectively in the scenes $\bar{\sigma}_t = \{\sigma_t[i]\}$. Moreover, we indicate with $\bar{\sigma}_{t+1} = \{\sigma_{t+1}[i]\}$ the vector of the scenes they face after the execution of $\bar{\alpha}_t$ and with $\bar{e}_{t+1} = \{e_{t+1}[i]\}$ the vector of expected situated actions at time $t+1$.

Let $\mathcal{L}$ be a language used to describe the environment (agents and objects), actions, scenes, situations, situated executed actions and expected situated actions. Let Σ_0 be *a priori* theory that describes the environment and the relations among agents and objects in term of actions, scenes, situations and situated executed actions.

Given two groups of agents G and G' (G,$G' \subseteq \mathcal{P}$), we define:

- *Cultural Constraint Theory for* G as a theory expressed in the language $\mathcal{L}$ that predicates on the expected situated actions of the members of G. If an expected situated action, estimated by the situated executed actions of G, satisfies the Cultural Constraint Theory, then the theory is said to be *validated*;
- a *Cultural Action w.r.t.* G is an executed action that satisfies the validated cultural constraint theory for G;
- *Implicit Culture phenomenon* when the members of G' execute cultural actions w.r.t. G without knowing the Cultural Constraint Theory for G.

Notice that G and G' can be in any relation, and just as a particular case they can coincide.

3.2 The architecture

The main goal of a SICS is to establish an implicit culture phenomenon. In the following we propose a general architecture that allows to achieve such a goal by:

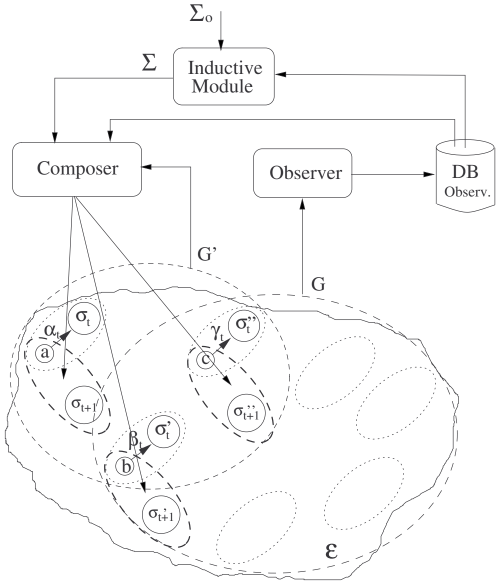

Figure 2: Architecture

- elaborating a validated cultural constraint theory Σ from a given domain theory and a set of executed situated actions executed by a group G;
- proposing to a group G' a set of scenes such that the expected situated actions of the set of agents G' satisfies Σ.

The architecture (Figure 2) consists of the following three basic components:

- *Observer* that stores in a data base (DB Observ.) the situated executed actions of the agents of G.
- *Inductive module* that, using the situated executed actions in DB Observ. and the domain theory Σ_0, induces a validated cultural constraint theory Σ;
- *Composer* that proposes to a group G' a set of scenes $\bar{\sigma}'_{t+1} \neq \bar{\sigma}_{t+1} = \{\sigma_t[i] = F_{\mathcal{E}}(\alpha_t[i], \sigma_t[i], t)\}$ such that their expected situated actions $\bar{e}_{t+1}$ satisfies Σ.

In Figure 2 the composer proposes to the agents a, b, and c the scenes σ_{t+1}, σ'_{t+1}, and σ''_{t+1}, respectively. Notice that in this case the agents b and c belong to both G and G'. This means that also their situated actions are stored in DB Observ. and thus they are used to elaborate the theory Σ and the new scenes.

3.3 Market example

Let us consider an environment *Market* in which there are a set $\mathcal{P}$ of agents (buyers and seller) and a set of objects $\mathcal{O}$ (books and money). Let $\mathcal{A}$ be a set of actions and $\mathcal{S}_{Market,\mathcal{A}}$ the scene space. In particular we consider the following actions:

- *ask(o,p)*: asking for the object o for the price p
- *offer(o,p)*: offering the object o for the price p
- *buy(x,o,p)*: buying from the agent x the object o at the price p
- *sell(x,o,p)*: selling from the agent x the object o at the price p

Let Σ_0 a priori domain theory that states that for any agents x and y, if at the time t, x is facing the scene σ_x (which contains y) and it asks for an object o for the price $p1$ and y is facing a scene σ_y (which contains x) and it offers the same object for the price $p2 > p1$, then at the time $t+1$ there exists a price $p3$ at which x buys from y the object o in a scene σ'_x and y sells to x the object o in a scene σ'_y. In other words Σ_0 states that the negotiation between buyer and seller must be successfully concluded in one step. We use for Σ_0 the following notation:

$$\begin{array}{l} \forall x, y \in \mathcal{P}, \\ \forall \sigma_x = \langle E_x, A_x \rangle, \sigma_y = \langle E_y, A_y \rangle \in \mathcal{S}_{Market,\mathcal{A}} : \\ \qquad \langle x, \sigma_x, t \rangle \wedge \langle x, ask(o, p1), t \rangle \wedge x \in E_y \wedge \\ \qquad \langle y, \sigma_y, t \rangle \wedge \langle y, offer(o, p2), t \rangle \wedge y \in E_x \rightarrow \\ \qquad \exists \sigma'_x, \sigma'_y \in \mathcal{S}_{Market,\mathcal{A}}, p3 : \\ \qquad \langle x, \sigma'_x, t+1 \rangle \wedge \langle x, buy(y, o, p3), t+1 \rangle \wedge \\ \qquad \langle y, \sigma'_y, t+1 \rangle \wedge \langle y, sell(x, o, p3), t+1 \rangle \end{array} \tag{3}$$

Let Σ a cultural constraint theory induced by the Inductive Module of the SICS using the situated executed actions of the agents of $G \subseteq \mathcal{P}$. Let suppose that Σ states that:

$$\begin{array}{l} \forall x, y \in G, \\ \forall \sigma_x = \langle E_x, A_x \rangle, \sigma_y = \langle E_y, A_y \rangle \in \mathcal{S}_{Market,\mathcal{A}} : \\ \qquad \langle x, \sigma_x, t \rangle \wedge \langle x, ask(o, p1), t \rangle \wedge x \in E_y \wedge \\ \qquad \langle y, \sigma_y, t \rangle \wedge \langle y, offer(o, p2), t \rangle \wedge y \in E_x \rightarrow \\ \qquad \exists \sigma'_x, \sigma'_y \in \mathcal{S}_{Market,\mathcal{A}} : \\ \qquad (E(h_{x,t+1} | \langle x, \sigma'_x, t+1 \rangle) = buy(y, o, p3)) \wedge \\ \qquad (E(h_{y,t+1} | \langle y, \sigma'_y, t+1 \rangle) = sell(x, o, p3)) \wedge \\ \qquad p3 = \frac{9}{10} p2. \end{array} \tag{4}$$

that is, for any agent x and y of G if x asks for an object o for the price $p1$ and y offers the same object for the price $p2 > p1$, then the expected situated actions for x and y are respectively of buying from y the object o and of selling to x the object o at $\frac{9}{10}p2$. Roughly speaking, this means that the buyers and the sellers of G usually agree on a 10% discount. Moreover, Σ says also that the negotiation between buyer and seller takes one step.

Let suppose now that at time $t = 1$ an agent $a \in \mathcal{P}$ asks for a *book* for \$100 and an agent $b \in G$ offers the *book* for \$200. In this case the Implicit Culture phenomenon is established if at the time $t = 2$ the agent a buys from b the *book* at \$180 without need to know that b usually makes a reduction in price of 10%.

In order to do this, at the time $t = 1$ the composer observes the two actions performed by a and b and using the situated executed actions in DB Observ. composes two scenes σ'_a and σ'_b, respectively for a and b, such that the expected situated actions for a and b satisfy the theory Σ. For instance, σ'_a and σ'_b could be two scenes in which a can ask for the *book* to \$180 and b can offer the *book* for \$180, and for which the expected situated actions are:

$$E(h_{a,2}|\langle a, \sigma'_a, 2\rangle) = buy(b, book, 180)$$
$$E(h_{b,2}|\langle b, \sigma'_b, 2\rangle) = sell(a, book, 180)$$

The implicit culture phenomenon is obtained if a buy from y the *book* at \$180, i.e. if a executes cultural actions w.r.t. G. Of course both a and b are always free to decide whether or not to buy and sell the *book*.

In this example the SICS is used as a mediator between two agents. Even if the mediation do not produce an agreement (i.e., a does not buy from b the *book* at \$180), it has avoided to the two agents to contract the price. The two agents can always start a negotiation, but now starting from \$180.

4. INSTANCES OF SYSTEMS FOR IMPLICIT CULTURE SUPPORT

A SICS based on our architecture enables an agent to perform a more effective behavior in a new environment. For instance, a SICS that intercepts the commands invoked by the users of a system can discover the printers that are used from a set of workstations, and predefine the aliases of a new user. Far from our simple example, instances of SICS can be found in component of existing systems. In particular, we show that a popular products reccomenders, a search engine, systems and a design support system has components that can be considered SICSs.

Collaborative Filtering (CF) [4] can be seen as an instance of our architecture. The goal of collaborative filtering is information filtering, namely to extract from a usually long list of items like links or products a little set that encounters the preferences of a user. Collaborative filtering reaches the goal exploiting the preferences, expressed actively o passively by other users in terms of ratings. Recommendations are built given the correlations between patterns of ratings on the items.

In this case the environment $\mathcal{E}$ is composed by items and ratings. The agents belonging to $\mathcal{P}$ are users. An agent can explicitly perform a rating action on an item *express(item1, rating1)* or some other actions like *choose(item1)* or *buy(item2)*, ... that the system assumes to be a rating by associating, for example, *buy(item1)* with *rating1*. We indicate with $\mathcal{A}$ the set of these actions. The *a priori* domain theory Σ_0 in the case of a collaborative filtering system is composed by:

$$\begin{aligned} &\forall x \in P, \exists \sigma_x : \forall \sigma'_x \neq \sigma_x \in \mathcal{S}_{\mathcal{E},A} \\ &\qquad E(h_{x,t}|\langle x, \sigma_x, t\rangle) = express(o, r) \wedge \\ &\qquad E(h_{x,t}|\langle x, \sigma'_x, t\rangle) = express(o', r') \rightarrow \\ &\qquad r' < r. \end{aligned} \tag{5}$$

where the scenes σ_x and σ'_x contain o and o' respectively and

$$\begin{aligned} &\forall \sigma \in \mathcal{S}_{\mathcal{E},\mathcal{A}},\ \exists K \subseteq G \subseteq \mathcal{P} : \\ &\qquad \forall x, y \in K\ (x \neq y),\ E(h_{x,t}|\langle x, \sigma, t\rangle) = E(h_{y,t}|\langle y, \sigma, t\rangle) \end{aligned} \tag{6}$$

The first term means that given an user there exists a scene

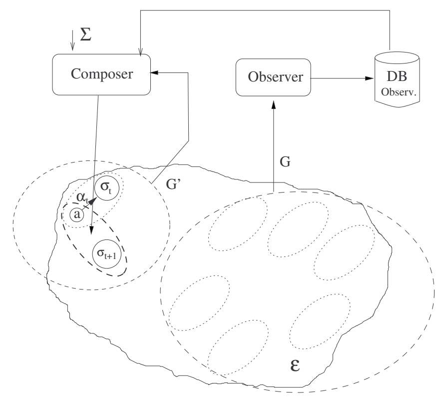

Figure 3: Memory–Based Collaborative Filtering as a particular case.

such that the rating associated with the expected situated action is a maximum with respect to the other possible scenes. The second term expresses the notion that the preferences of the users cluster.

In the case of model-based Collaborative Filtering the Inductive Module characterizes the sets K depending on the situated executed actions and adds the characterizations to Σ. Obviously, collaborative filtering algorithms express the characterization in a non-logical form and sometimes even not in a explicit way. Figure 3 shows the architecture in the particular case of memory–based collaborative filtering where no theory is explicitly built. The theory Σ_0 is directly inserted into the composer.

Our architecture covers collaborative filtering as a particular case. That means that collaborative filtering establishes an implicit culture phenomenon. Leug has already noted that collaborative filtering changes the social nature of recommendation [7].

A rather popular application of collaborative filtering is exploited in the site amazon.com. In this case the system uses information about book orders of past customers to suggest relevant products when a user is browsing the site. Related to collaborative filtering is the DirectHit[1] technology for search engines used in popular sites such as lycos.com and hotbot.com. The search engine intercepts the choices of the bookmarks done by the users, given a set of keywords, and use this information for changing the ranking of bookmarks on future similar searches. The performance of a user is improved by the knowledge of other users in a perfectly transparent way. Finally, the Stamping Advisor system reported by Leake et al. [6] uses a Case Based Reasoning engine in order to provide useful information for supporting stamping design activity in car manufacture. The information is provided pro-actively with a "just-in-time retrieval" without any need of request by the user and the cases are collected as a by-product of user's decision making. The system maps to our architecture because the Inductive module is realized by a CBR engine, and the observations of scene and actions does not interfere with the activities of the users.

[1] *http://www.directhit.com/about/products/technology_whitepaper.html*

Our approach generalizes the instances in different directions. First we pose the Implicit Culture phenomenon in an agents framework and gives the premises for exploit it also for artificial agents. Second, we generalize the forms of Cultural Constraints. Finally, the general form of SICS support a group of agents in an integrated way and not only one by one.

5. RELATED WORK

Despite its centrality in *Cultural Anthropology* the notion of culture resisted several attempts of definition. Following the most accepted definitions, the concept of culture covers almost all the activities that a group of humans does on a particular geographic area, as well as its material or symbolic production. Obviously, we do not try to address the complete and complex cultural processes of a group of agents but we limited our attention to actions and behaviors. To this regard (i.e.,to observe the behavior in order to provide a support) our approach is more related to the use of ethnographic methods for requirements specification [8] rather than to Anthropology *tout court*.

In Artificial Intelligence there have been already some attempt to address cultural issues. Proposed by Reynolds et al. *Cultural Algorithms* [9] concentrate on the aspect of shared knowledge of cultural phenomena. Strongly related to genetic algorithms, *Memetic Algorithms* (see for example [3]) address the problem of evolution of culture in terms of evolution of ideas. The ideas, no matter which is their support, interacts one another and the interaction, via mutation or cross-over phenomena generates different ideas. Finally, there have been the proposal of *Artificial Culture* [2] that can be seen as the natural evolution of the approach of Artificial Life combined with Multi-Agents Systems. The goal is to simulate cultural evolution in an environment in which groups of agents exchange products and communicate one another.

The main difference with the above approaches is that we are not trying to reproduce a generic cultural phenomena but only a cultural behavior. Moreover, our main issue is not simulative but rather to individuate an effective architecture for improving agent-based systems.

Our architecture is also related to work done in the *Adaptive Interfaces* area and to the notion of situated action that has a long history [10] and has originated strong debates [5]. The user-interface of a system is dynamically changed and the different presentation is guided by the interaction history. We have already shown in Section 4 how *collaborative filtering* is an instance of the architecture. A strong correlation is also present with the wide area of *User Modeling*. User modeling deals with prototypized users' profiles that are assigned with a user classification process (for an application in e-commerce see for instance [1]). Rather differently, our approach do not require to classify agents nor building abstract profiles of their interactions. Our contribute is to emphasize the importance of putting into a relation the behaviors of different agents without requiring an explicit effort from them.

6. CONCLUSIONS

We have introduced and defined the notion of Implicit Culture phenomenon showing how it can be useful in order to help agents to act effectively in an environment. We have presented a general architecture for Systems for Implicit Culture Support, namely systems aimed to establish an implicit culture phenomenon on a group of agents. The architecture covers Collaborative Filtering as an instance and also suggests further applications with human and artificial agents. The main advantage of SICS is that they are completely external to the agents and it can boost their activities and effectiveness without requiring additional computational load.

7. ACKNOWLEDGMENTS

The authors wish to thank Fausto Giunchiglia for precious insights and continuous support and Paolo Avesani, Alessandro Ebranati, Sabrina Recla and Luciano Serafini for useful discussions.

8. REFERENCES

[1] L. Ardissono and A. Goy. Tailoring the interaction with users in electronic shops. In Judy Kay, editor, *User Modeling: Proceedings of the Seventh International Conference, UM99*, pages 35–44. Springer, Wien New York, 1999.

[2] Nicholas Gessler. Artificial culture. In Rodney Brooks and Pattie Maes, editors, *Fourth International Workshop on the Synthesis and Simulation of Living Systems*, pages 430–435, Cambridge, 1994. MIT Press.

[3] D. Hales. Selfish memes and selfless agents - altruism in the swapshop. In *Third International Conference on Multi-Agent Systems (ICMAS98)*. IEEE Computer Society, 1998.

[4] J. Herlocker, J. Konstan, A. Borchers, and J Riedl. An algorithmic framework for performing collaborative filtering. In *Conference on Research and Development in Information Retrieval*, 1999.

[5] N. Kushmerick. Cognitivism and situated action: two views on intelligent agency. *Computers and Artificial Intelligence*, 15(5), 1996.

[6] D. B Leake, L. Birnbaum, K. Hammond, C. Marlow, and H. Yang. Integrating information resources: a case study on engineering design support. In Klaus-Dieter Althoff, Ralph Bergmann, and L. Karl Branting, editors, *Case–Based Reasoning Research and Development*. Springer, Berlin, 1999.

[7] Christopher Lueg. Considering collaborative filtering as groupware: Experiences and lessons learned. In *Proceedings of the Second International Conference on Practical Aspects of Knowledge Management (PAKM'98)*, Basel, Switzerland, 1998.

[8] Bonnie A. Nardi. The use of ethnographic methods in design and evaluation. In M. Helander, T. K. Landauer, and P. Prabhu, editors, *Handbook of Human-Computer Interaction*, pages 361–366. Elsevier Science, 1997.

[9] Robert G. Reynolds. An introduction to cultural algorithms. In *Proceedings of Evolutionary Programming (EP-94)*, San Diego, CA, 1994.

[10] Lucy A. Suchman. *Plans and Situated Action.* Cambridge University Press, 1987.