How to Integrate Personalization and Trust in an Agent Network

Laurent Lacomme[1], Yves Demazeau[1], and Valérie Camps[2]

[1] Laboratoire d'Informatique de Grenoble, Grenoble, France
[2] Institut de Recherche en Informatique de Toulouse, Toulouse, France
{Laurent.Lacomme,Yves.Demazeau}@imag.fr,Valerie.Camps@irit.fr

Abstract. Trust and personalization are two important notions in social network that have been intensively developed in multi-agent systems during the last years. But there is few works about integrating these notions in the same network of agents. In this paper, we present a way to integrate trust and personalization in an agent network by adding a new dimension to the calculus of trust in the model of Falcone and Castelfranchi, which we will call a similarity degree. We first present the fundamental notions and models we use, then the model of integration we developed and finally the experiments we made to validate our model.

Keywords: Trust, Agent network, Personalization, Social network.

1 Introduction

From Web Services to experimental negotiating agendas, many multi-agent systems have been developed to implement links between people or organizations in order to enable them to interact indirectly through agents that represent them. In such *social networks*, each agent stands for a person or a group of people.

These networks have often some particular properties. The first we are interested in is *openness*. In open networks, agents can be added or removed from the network at any time. This implies that the network evolves, while each agent needs to adapt its own behavior to the appearance or disappearance of partners. The second property is *partial representation*. In social networks, agents often only have little knowledge about others and about the network itself. For instance, when an agent is added to the network, it usually only knows a few other agents that we call its neighborhood. A third interesting property is *heterogeneity*. That is, in such networks, agents are not always homogeneous. Every agent can have individual skills that others do not have, and each agent is free to cooperate or not with known agents. So each agent has to choose cleverly its partners in this kind of networks, because these partners must fulfill some requirements for the partnership to be useful.

Hence such networks need some protocols for the agents to be able to act correctly while knowing only a few facts about a constantly evolving environment. One way to fulfill this requirement is to add a trust model to their reasoning abilities. This trust model enables them to take decisions, such as which agent to ask for doing a task, from the little knowledge they have. This is done by first computing probable

J. Filipe, A. Fred, and B. Sharp (Eds.): ICAART 2009, CCIS 67, pp. 247–259, 2010.

behaviors of others and the results of such behaviors, and then selecting the best ones for the agent.

On another hand, as the agents in these networks are used to represent human beings, users often want to have some control over them. Indeed, when agents are faced to choices, their reactions should be the closest possible to the users' own preference. One way to realize it is to add to the agents' reasoning methods a personalization model, which checks for alternatives and selects the one the user would prefer.

As a result of the two previous remarks, there is a need to include both trust and personalization models. But as both are reasoning methods that can lead to contradictory conclusions, we need a way to integrate them in the agent's global reasoning protocol. From our best knowledge, such an integration does not seem to exist yet.

In the remaining of this paper, we will first present the notions of trust and personalization in a network of agents, the theoretical criteria we will develop on these notions for our integration work and the trust model we chose as a foundation to our integration model. Then, we will describe the integration criteria and the solution we are proposing. And finally, we will present an experimental validation of our solution.

2 Positioning

There are many ways for agents to represent and compute trust they have in other ones, and there are also many ways for them to represent and exploit user's preferences. So, to understand how trust and personalization should interact in an agent's reasoning schema, we firstly need to describe what they are, how they work, and what their different possible models are.

2.1 Trust in Agent Networks

A trust model describes how an agent can use its past experience and others' experience to take decisions about future plans. It involves a facts storing and a reasoning method over this memory. As we are interested in user-representing agents, the most common way to describe agents' reasoning methods is through the *Beliefs, Desires, and Intentions* – BDI – paradigm (Rao & Georgeff, 1995).

Trust is based on trust evidences (Melaye & Demazeau, 2006), which are facts that are relevant to the question of trusting an agent or not, which can come from different sources. Common trust sources are: *direct experience*, which can be positive or negative, *reputation*, which is an evaluation that a third-party agent provides about another one, and *systemic trust*, which is the trust an agent has in a group of other agents, without necessarily knowing specifically each member of the group. These evidences are stored in a way so that they can later be used by a reasoning process, *i.e.* beliefs in the case of BDI agents. Moreover, all trust knowledge is contextual, *i.e.* it is related to an action or a goal Ω the agent wants to perform or achieve.

Most commonly, these beliefs are split into a small number of categories that are considered as trust dimensions. The most used dimensions can be described as *ability*, *willingness* and *dependence* beliefs (Castelfranchi & Falcone, 1998). The belief of *ability* means that an agent A believes that an agent B is able to do what A wants it to do in the context Ω. The belief of *willingness* means that A believes that B will do what A wants it to do if A asks it to do that. The belief of *dependence* means that A believes that he relies on B to achieve its goals in context Ω. There exist other dimensions, but trust can often be easily simplified to retain only these three ones without losing any accuracy.

Trust is then learnt through experience, interaction and reputation transmission, stored as agents' beliefs and then used in decision-making processes when agents have to make choices involving other agents.

Amongst the large amount of existing trust models, we need to rely on some criteria to make our choice and ground our work on an adequate model.

2.1.1 Some Theoretical Criteria for Trust Models

A trust model has to fulfill some criteria to be able to be used in an agent network to create what is usually called a *trust network*.

The first obvious criterion is *optimization*. This comes from the fact that a trust model is made to help agents to adapt their behavior to the network configuration. So the trust model must be able to improve the network's global performance – the ability for each agent to achieve its goals. *Optimization* can only be tested experimentally, because we are not able to foresee the network improvement given a particular theoretical trust model.

The second one is a practical requisite: the trust model should be easily calculable, in order for the agent to compute it in real-time without any significant lack of reactivity; we call it *calculability*.

The third one is related to the fact that the agents we describe are related to users. In many of these systems, users often want to be able to understand how the entity that represents them reacts. So, the *intelligibility* criterion describes the ability of the reasoning process and the semantic of stored beliefs to be explained to the user and understood by him.

The other criteria we take care about are four properties of the trust values (Melaye & Demazeau, 2006): *observability, understandability, handlability* and *social exploitability*. They describe the ability of the agent to apprehend other agents' mutual trust, to compute multi-dimensional trust values, to combine these values into a global trust level and to use this trust level to make decisions.

We will use these criteria altogether for both the choice of the trust model to ground our integration work and, later, for the integration model itself.

2.1.2 Falcone and Castelfranchi Trust Model

The trust model we used is the one introduced by Falcone et al. (Castelfranchi & Falcone, 2004). It is a BDI-based model – which corresponds perfectly to the social network requisites – and uses numerical representation for trust beliefs. Numerical representations are better for a trust network than logical representations, because these latest rely quite always on complicated and high-complexity modal logics, and so cannot be implemented.

This model uses three contextual values to represent trust dimensions (cf. figure 1): the *Degree of Ability* ($DoA_{Y,\Omega}$), the *Degree of Willingness* ($DoW_{Y,\Omega}$) and the *Environment Reliability* ($e(\Omega)$). The latest is the only one that does not depend on the agent B that is considered by A for the trust evaluation. It measures the intrinsic risk of failure due to the environment. The *Degree of Ability* measures the competence that B has to

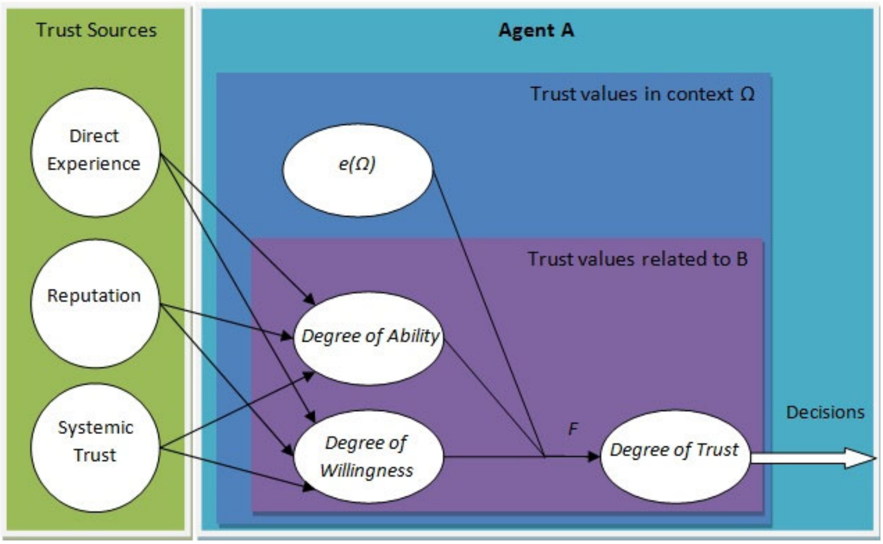

Fig. 1. Falcone and Castelfranchi Trust Model

accomplish the task or to help A to accomplish the task Ω. The *Degree of Willingness* measures the will of B to help A to achieve its goals. All these three values are defined in [0;1] and are combined into a global *Degree of Trust* ($DoT_{Y,\Omega}$) that describes the trust that A has in B in context Ω through a function F. This function is not specified in the model, but has to preserve monotonicity and to range in [0;1].

$$DoT_{Y,\Omega} = \mathrm{F}(DoA_{Y,\Omega}, DoW_{Y,\Omega}, e(\Omega)) \tag{1}$$

Both of the ability and willingness beliefs – *DoA* and *DoW* values – can be learnt from any trust source. To learn these values, the agent uses a reinforcement learning process that uses new trust evidences to review its knowledge about others.

We chose this model because it satisfies very well the criteria we have for a trust model. The fact that trust is computed from values representing its dimensions with a simple formula guarantees the respect of *observability, understandability, handlability* and *intelligibility. Social exploitability* is also respected because trust values provide a ranking for potential partners that can be used to make a decision. And then, this model is *calculable*, because it is based on simple mathematical formula and numerical values.

2.2 Agents Personalization

The other fundamental notion in this work is personalization. Personalization is the ability for an agent acting on behalf of a user to acquire and to learn his preferences, his centre of interests and to use them during its decision-making process.

While preferences nature is quite domain-related, preferences representation has some universal methods and properties.

2.2.1 Preferences Representation Models

The notion of personalization handles a couple of distinct concepts. It is both a way to represent users' preferences, a way to learn them from the user and a method to use them in various contexts to improve the agent's behavior to the user's point of view.

There are two distinct ways to represent users' preferences (Endriss, 2006). They can be represented by a valuation function giving a note to alternatives the agent is faced to – and called *cardinal preferences*. They can also be described as a binary relation between each two of the alternatives – and called *ordinal preferences*.

There are many ways to represent these two kinds of preferences. But the most known and useful are probably the weighted conjunction of literals for *cardinal preferences* and the prioritized goals for *ordinal preferences*. These models describe a way to store user preferences but also a way to use them in the reasoning process by evaluating and choosing one between several alternatives.

All these preferences representation models can be combined with several well-known reinforcement learning techniques (Gauch et al, 2007), which will enable them to improve the precision of the user profile (*i.e.* the set of all represented user's preferences) and adjust it to the user's real preferences. The learning process can use an *explicit feedback*, which can be, for instance, a form that the agent presents to the user. It can use an *implicit feedback*, which is the analysis of the user activity, for instance, the user web history for web navigation assistants. Or it can use *hybrid feedback*, which is a combination of both (Montaner et al, 2003).

2.2.2 Some Criteria for Personalization

As for trust models, we proposed a set of important criteria that a personalization model has to respect in order to be useful in an agent network.

In our work, we have kept four usual criteria about personalization models (Endriss, 2006) and added a new one. Firstly, the *expressive power* is defined as the amount of preferences structures that the model is able to represent. Secondly, *succinctness* is the amount of information about these preferences which can be stored in a given place. Then, we have to take care about *elicitation*, which represent the ease with which a user can formulate his preferences in the model's representation language. This is an important criterion, especially when a user is able to see directly his profile and to modify or correct it on his own. And finally, as it is also the case for trust model, the *complexity* of the model is important in order to be able to be computed in real-time by agents.

Since the preferences learning mechanism is a dynamic process, we have to describe the ability of the system to react to any change in the user's preferences. This is why we add another criterion, *reactivity,* which measures how much time the model takes to adapt the profile to a change in the user's behavior it represents.

3 Integration Work

Concepts of trust and personalization having been studied in a network of agents, we will now see why there is a necessity to find a way to integrate these two reasoning processes into a single one.

3.1 Personalization Integration in a Trust Network

In a lot of networks – agent-based social networks, B2B applications and negotiating calendars for instance – the agents have to make decisions about which other agents to interact with, which ones to ask information to, and with which to form teams or to take contracts and partnerships. Agents often take these decisions via a trust mechanism, but in such networks, the user is often able to express preferences that should influence these choices. So, as trust and personalization needs to coexist in those networks, we obviously need a way to make them function altogether.

3.1.1 The Necessity of Integration

We can first believe that simply putting both models on the same agents will be enough to make the network work well. But this cannot be true, because both being reasoning processes that cost much time and resources and that leads to conclusions, there will be two main problems happening. The first one is that the cost of both inferences will be high for an agent. The second one, and most important, is that the two inferences can lead to different and perhaps incompatible conclusions. And the problem will be: how to handle these two conclusions and act while taking both into account.

So the way to solve this problem is to create a single reasoning process for agents, that takes into account all the knowledge they possess, about both the network (other agents) and user's preferences.

3.1.2 Related Work

The only work we have found in the literature that tries to integrate an approach of personalization and trust (Maximilien & Singh, 2005) proposes a model of multi-criteria trust in which the user has some control over the importance of each evaluation following a particular criterion in the final trust calculus. This is a very limited and particular sort of personalization, and this approach is not applicable for the kind of networks we are interested in, as the preferences are related to the trust model itself and not to the domain the agents are concerned with.

So, to the best of our knowledge, no work exists that tackles the interaction of these two notions in an agent in the way of providing a single reasoning process that handles both notions.

3.2 Considered Agents and Network

To be able to explain clearly how the solution we propose works, we first need to describe the agents and the types of networks in which it will be applied.

3.2.1 Agents' Architecture and Capacities

The agents are based on the BDI architecture. This means that (*i*) they possess some *beliefs* about their environment – including the users of the system – and other agents and (*ii*) that they all have some goals, called *desires*, which are states – personal or of the environment – they wish to be true. In order to make these goals true, (*iii*) they use plans to make decisions that become *intentions* – things they plan to do.

As every agent does not have all the ability needed to achieve every one of its goals, it has to cooperate with some other agents in the network. To minimize the cost of this required cooperation and to avoid losing time and resources asking wrong agents for help – wrong agents are those which cannot help or will not help – it uses some trust process to determine which agents are the best partners for a specific task by the mean of the previously described Falcone and Castelfranchi trust model.

Every agent should also be related to a single user, and should be able to stock and use a preference profile related to this user. We will see later how the agents are able to do that.

3.2.2 The Network Structure

The network is merely an evolving set of agents which are able to communicate one with another through a message protocol that enables them to exchange data, requests, answers and perhaps beliefs and plans.

When a new agent is added to the network, it knows a few other agents – its neighborhood – and can learn the knowledge of other agents by interacting with them. It order to make agents able to learn the existence of unknown other agents, we must include in the network a mechanism that makes agents who are not able to process an information or answer a request forward this request to another agent. This mechanism involves only agents that are not concerned by a request or information but are cooperative – they are ready to help the request sender or to distribute information in the network. When such an agent receives an irrelevant message from its viewpoint it does not ignore it but forwards it to one or several agents of its own neighborhood which it considers as the most able to answer to this message; this mechanism is called *restricted relaxation* (Camps & Gleizes, 1995). The number of times a message can be forwarded in the entire network is obviously limited to avoid cycles and thus network overload.

3.3 Integration Constraints

As for trust and personalization models, we propose some theoretical criteria needed to be fulfilled by an integration model.

The first criterion derives directly from the fact that we want to integrate a personalization model in a trust model: this operation needs to result in an improvement of the correlation between the user's expectations and the agent's behavior. Hence the criterion of *accuracy* will be the measure of proximity between user's desires and agent's observable behavior and results.

Two other criteria will simply describe facts that are linked with the operation we are trying to do. The *target correlation* describes the fact that the alternatives considered by the preferences profiles are defined by the integration model we choose. In other words, the personalization model must be able to evaluate the kind of alternatives that the trust model will require it to evaluate. On the other hand, the *type correlation* criterion describes the fact that the personalization model should give as a result of an alternative evaluation a value of a type that is useful to the trust model.

Finally, because both of the two models we are trying to integrate are contextual models, we have to be sure that the contexts defined in both of the models are compatible – which means that they are the same or at least one is a subdivision of the other. We will call it the *context compatibility*.

3.4 Towards an Integration Model

Taking into account all the listed criteria, and basing our work on the chosen trust model, we developed a solution for an integration of a personalization model into this trust model. This integration has been done in order to create a reasoning process that handles information about both user's preferences and other agents' behavior.

3.4.1 Description of the Proposed Integration Model

The solution we propose simply consists in defining a new dimension of trust in an already multi-dimensional trust model (cf. figure 2). Indeed, in order to take into account the personalization evaluations, we considered that it was quite the best solution to keep the preferences representations and learning methods as separate agent's ability. We made this choice because it seemed very difficult and confusing – both for programmers and users – to incorporate it to the trust model reasoning process.

So this process is only going to use the evaluator from the personalization model to rank alternatives between other agents' behaviors or results. It is also going to learn a new context-dependant, agent-dependant belief that represents the proximity this

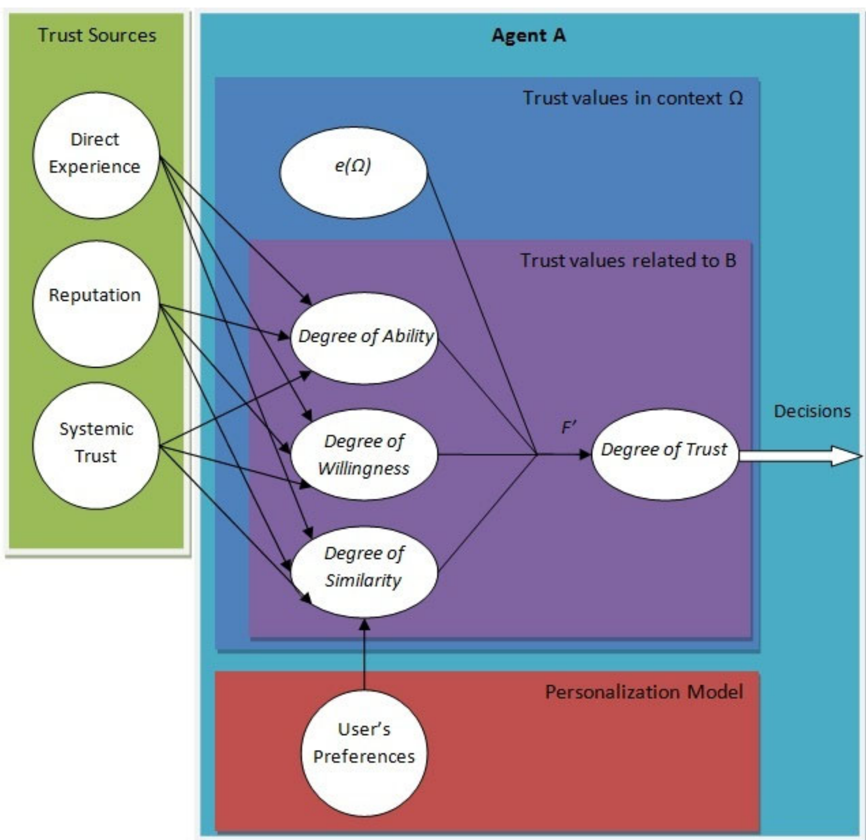

Fig. 2. Modified trust model with integrated personalization model

agent's behaviors or results have with the user's expectations concerning a particular context.

This belief, defined as a numerical value ranging in [0;1], as other trust values, is also going to be learnt by reinforcement learning methods using personalization model's evaluators. This will be called the *Degree of Similarity* $DoS_{Y,\Omega}$. Then, the global trust computation will have to be redefined as another monotonic function F' that ranges in [0;1], which takes as parameters not only the parameters of the function F from the trust model but also the newly defined *Degree of Similarity* :

$$DoT_{Y,\Omega} = F'(DoA_{Y,\Omega}, DoW_{Y,\Omega}, DoS_{Y,\Omega}, e(\Omega)) \tag{2}$$

Moreover, this new dimension of trust can be learnt from every trust source that is available in the trust model we use. However, to learn it from reputation an agent will have to take into account the similarity level between itself and the evaluator, as they do not have the same user profiles.

The constraints that this integration model makes on the choice of a preferences representation model can be explained through the *type correlation* and *target correlation* criteria: the Falcone and Castelfranchi trust model uses numerical values in [0;1] as trust dimensions values. So, as our new belief will also have to be represented as such a numerical value, and because, for learning, the evaluatedelements will be the results of one interaction with another agent, the personalization model's evaluation functions will have to evaluate a single result and give a numerical value as an evaluation. So the personalization model should be a *cardinal preferences* model.

3.4.2 Criteria Applied to the Proposed Model

As previously shown, the original trust model we used fulfills all required criteria. So, as we have just added a single trust dimension, with its own meaning and its own learning methods, these criteria will not be broken. The *intelligibility* will also be respected, because the meaning of this dimension is easily explainable to the user; it represents the proximity between the real behavior of the target agent and the theoretical behavior it should have, taking into account the user's preferences. In fact, only the *optimization* criterion needs to be experimentally tested.

The criteria related to the personalization model could easily be satisfied, because the choice of the model is quite free between all the cardinal preferences models. They can be satisfied, for instance, by choosing the weighed conjunctions of literals model, which is a light, low-complexity and powerful model which can handle every domain of application and is perfectly compatible – if correctly implemented – with the contexts of the trust model.

Amongst the integration criteria, the two *correlation* criteria are easily respected, as seen before. The *context compatibility* can also be satisfied by correctly implementing the personalization evaluators. So, only the *accuracy* criterion should be experimentally tested.

4 Tests and Experiments

In order to first experiment the integrated trust and personalization model we proposed and then to check the two criteria that we can only validate experimentally – optimization and accuracy – we implemented a simplified version of the model, using the agent programming language and IDE Jack (http://www.agent-software.com/shared/products/).

4.1 A Simplified Model

We firstly simplified the model to test only the two experimental criteria we exposed – the goal of the test was not to determine the efficiency of the Falcone and Castelfranchi trust model nor of any personalization model, but to experiment if the integration model itself is viable.

The first simplification we decided was the implementation of a mono-source trust model. The only source that we considered was direct experiment. So the only trust evidences that were taken into account by the trust learning process were answers given by other agents to the sent requests.

The second simplification we decided concerns the environment; it was supposed to be sure – every message reaches its addressee – and resourceful – if an agent has both the ability and the willingness to perform an action, then the action is performed. So we considerer $e(\Omega)=1$.

We also selected simple functions for trust computation and for learning. F' (for complete model) and F (for trust only model) are defined as simple multiplications between each dimension.

$$DoT_{B,\Omega} = \mathrm{F'}(DoA_{B,\Omega}\,,\, DoW_{B,\Omega}\,,\, DoS_{B,\Omega}) = DoA_{B,\Omega}\ \mathrm{x}\ DoW_{B,\Omega}\ \mathrm{x}\ DoS_{B,\Omega} \tag{3}$$

$$DoT_{B,\Omega} = \mathrm{F}(DoA_{B,\Omega}\,,\, DoW_{B,\Omega}) = DoA_{B,\Omega}\ \mathrm{x}\ DoW_{B,\Omega} \tag{4}$$

Learning method is defined as a weighted mean of current and new values with fixed ratios – let *DoX* be *DoA* or *DoW, and* $a,d\in [0,1]$.

For positive trust evidence:

$$DoX(t+1) = DoX(t) + a * (1-DoX(t)) \tag{5}$$

For negative trust evidence:

$$DoX(t+1) = DoX(t) - d * DoX(t) \tag{6}$$

And for the similarity value – in case of positive or negative trust evidences:

$$DoS\,(t+1) = (DoS\,(t) + mod_{DoS}(B,\Omega))/2 \tag{7}$$

Finally, we decided to use a static representation of preferences described by a very simple user profile that enables an agent to rate every result it receives in [0;1]. We emulated a preferences model in that way, because preferences dynamics was not very important for these tests and, given the high number of possible preferences representations, this would not be significant anyway. So we faked a preferences model that would have reached a stable state by attributing a simple static profile to every agent.

4.2 The Experimental Protocol

We experimented in a network of 100 homogeneous agents able to possess 3 basic capacities A, B and C. Each newly created agent randomly receives the ability to use each one of the 3 capacities with a certain probability – p = 0.6 for most of the tests.

All the agents are able to communicate through a message protocol defined in the Jack interface, and all use *limited relaxation* paradigm – with a maximum of 5 successive relaxations for a message.

We used randomly generated initial neighborhood for each agent with a probability of knowing each other agents equal to 0.1.

Then, the process continues step by step. At each step, a goal is generated for each agent, which consists in using a random capability: A, B or C. Obviously, when the agent does not possess this particular capability, it has to cooperate with other agents to achieve its goal.

To emulate preferences evaluation, each result of a capability A, B or C, is a document, which is assessable by the personalization model of any agent, according to a simple user profile it possesses. So, when an agent uses one of its capabilities or gets a result from another agent, it is able to rate this result according to its own preferences profile. We thus defined $mod_{DoS}(B,\Omega)$ as the average pertinence of documents given to A by B as an answer to a request from A.

4.3 Experimental Results

We evaluated the number of messages exchanged between agents and the number of goals that were not achieved by agents to validate the *optimization* criterion. We also evaluated the average pertinence of results for the *accuracy* criterion.

Each value is measured at step 1 and step 100 for 4 different networks: (*i*) a simple network without any model, (*ii*) a network with the simplified trust model – without personalization –, (*iii*) a network with the trust and personalization integrated model, and (*iv*) a network with a model that only takes into account the personalization value. Then the results between step 100 and 1 are compared to measure improvement. Expected results (cf. table 1) are an increase for pertinence and a decrease for the two other values.

The results fit with our expectations: the number of messages and the number of failures decrease for all networks where there exist trust models, and the average pertinence of results increases significantly in networks where personalization is taken into account.

Table 1. Experimental results summary: evolution between steps 1 and 100

	Number of messages	Number of goal failures	Average pertinence
No trust nor personalization	-3.6%	+1.6%	-3.3%
Trust model only	-13.7%	-77%	+3.5%
Trust and personalization model	**-21.5%**	**-76%**	+11.9%
Personalization only	+34.3%	+27%	**+42.3%**

Complementary observations can be made; for instance, while in much cases network *optimization* is the same for trust only and for trust and personalization networks, we can observe that when the pertinence results are too often very low, the network obtains worse results, because of the unbalanced importance of the different trust dimensions; we can notice that this effect should probably be corrected by adjusting correctly the global trust computation function.

But, globally, we can conclude from these experiments that the two experimental criteria that we had expressed are satisfied, and so, that our model seems to be an adequate solution to the problem we wanted to address.

5 Conclusions and Perspectives

In this work, we have explored the possibility of associating trust and personalization paradigms in an agent network. We have done this in order to give to agents the ability to handle both the intrinsic uncertainty of a partial-knowledge, evolving network, and the also evolving requirements of a user's set of preferences. Indeed, agents would have to face them both in a social network in which each user has one or more agent to represent him.

Knowing that just putting together the two reasoning methods leads to heavy problems of optimization but also to problems for mixing the results given by each one, we have looked for a solution to integrate both notions in a single reasoning process. We first gave some theoretical criteria to choose every component of a global agent's reasoning method that could handle both trust and personalization: the trust model, the personalization model and the integration model. We then proposed a complete solution that is acceptable according to those criteria.

Our solution involves the Falcone and Castelfranchi trust model, to which we added a new trust dimension, that we called *degree of similarity*. It also involves a cardinal preferences model such as weighted conjunction of literals, which is used by agents to evaluate results and alternatives and learn the *degree of similarity* they have with other ones.

The obtained experimental results for *optimization* and *accuracy* criteria seemed to validate these criteria. That is why even if these experimentations were done on a simplified version of the trust and personalization integration model, we can say that the solution we proposed seems to be viable and to be applicable to the kind of networks we described. This model was developed in order to improve the behavior of agents in these open, partial-knowledge and user centered networks, and it seems to achieve this goal.

Future work on this solution is to test it with a full and multi-source implementation with dynamic personalization from real users. As the Falcone and Castelfranchi model is very powerful and because of the large scale of different cardinal preference implementations that can fit in the theoretical criteria of this solution, we can foresee very different solutions for various domains and the need to find the adequate personalization evaluation and trust evaluation functions to each model.

References

1. Camps, V., Gleizes, M.-P.: Principes et évaluation d'une méthode d'auto-organisation. In: 3èmes Journées Francophones IAD & SMA, St Baldoph, pp. 337–348 (1995)
2. Castelfranchi, C., Falcone, R.: Principles of trust for mas: cognitive anatomy, social importance, and quantification. In: 3rd Int. Conf. on Multi-Agent Systems, ICMAS 1998, Paris, pp. 72–79 (1998)
3. Castelfranchi, C., Falcone, R.: Trust dynamics: How trust is influenced by direct experiences and by trust itself. In: 3rd Int. J. Conf. on Autonomous Agents and Multiagent Systems, AAMAS 2004, New-York, vol. 2 (2004)
4. Endriss, U.: Preference Representation in Combinatorial Domains. Institute for Logic, Language and Computation, Univ. of Amsterdam (2006)
5. Gauch, S., Speretta, M., Chandramouli, A., Micarelli, A.: User Profiles for Personalized Information Access. The Adaptive Web, 54–89 (2007)
6. Maximilien, E.M., Singh, M.P.: Agent-Based Trust Model Involving Multiple Qualities. In: 4th Int. J. Conf. on Autonomous Agents and Multiagent Systems, AAMAS 2005, Utrecht (2005)
7. Melaye, D., Demazeau, Y., Bouron, T.: Which Adequate Trust Model for Trust Networks? In: 3rd IFIP Conference on Artificial Intelligence Applications and Innovations, AIAI 2006, Athens. IFIP (2006)
8. Montaner, M., López, B., De La Rosa, J.L.: A Taxonomy of Recommender Agents on the Internet. Artificial Intelligence Review 19, 285–330 (2003)
9. Rao, A., Georgeff, M.: BDI Agents: From theory to practice, Tech. Rep. 56, Australian AI Institute, Melbourne (1995)