Trust Based Evaluation of Wikipedia's Contributors

Yann Krupa[1], Laurent Vercouter[1], Jomi F. Hübner[1,3], and Andreas Herzig[2]

[1] École Nationale Supérieure des Mines de Saint Etienne
Centre G2I, Département SMA
158, cours Fauriel, F-42023 Saint-Etienne
{krupa,vercouter,hubner}@emse.fr
[2] Institut de Recherche en Informatique de Toulouse
118 route de Narbonne, F-31062 Toulouse
herzig@irit.fr
[3] Federal University of Santa Catarina
Department of Automation and Systems Engineering
P.O. Box 476, Florianópolis, Brasil, 88040-900
jomi@das.ufsc.br

Abstract. Wikipedia is an encyclopedia on which anybody can change its content. Some users, self-proclaimed "patrollers", regularly check recent changes in order to delete or correct those which are ruining articles integrity. The huge quantity of updates leads some articles to remain polluted a certain time before being corrected. In this work, we show how a multiagent trust model can help patrollers in their task of controlling the Wikipedia. To direct the patrollers verification towards suspicious contributors, our work relies on a formalisation of Castelfranchi & Falcone's social trust theory to assist them by representing their trust model in a cognitive way.

1 Introduction

In evolutive and open systems, like participative websites as Wikipedia[4], common security measures are too complex to be applied while maintaining a good tradeoff between openness and website integrity. For example, a Role Based Access Control (RBAC) limits the system openness thus reducing participation. Moreover, role management represents a heavy administrative task. Then, it is necessary to define decentralized control mechanisms allowing scalability. A multiagent approach of this problem is therefore appropriate, in particular by means of social control via trust assessment towards other agents of the system.

The work presented here defines how a multiagent trust model, inspired by the Theory of Social Trust by Castelfranchi & Falcone [1], can be constructed

[4] http://en.wikipedia.org/

to contribute to an application like Wikipedia. The content of Wikipedia is controlled by the users themselves. These users handle the modification checking, nevertheless, a few mistakes remain online for some certain time because of the sizeable activity of Wikipedia. The cognitive trust model that we propose allows to target more efficiently the contributions to check in priority by making trust assessments towards other users.

The structure of this article is the following: section 2 focuses on some of Wikipedia intrinsic mechanisms and particularly on patrollers who watch modifications. Section 3 is about trust in multiagent systems and describes the trust model used in our application. We apply this model to Wikipedia in section 4. Finally, we focus on the use of the trust model to maintain the integrity of the encyclopedia, by means of patrollers assistance, in section 5.

2 The Wikipedia Encyclopedia

The online encyclopedia Wikipedia is now famous and commonly used, it is one of the 10 most visited websites, according to the web traffic measurement website Alexa[5]. Wikipedia basic principle is that anyone can modify articles. They are constantly updated, corrected and completed. By submitting a modification, a user accepts that everything goes under free licence allowing free use and modification of the content. Those two principles make Wikipedia a free encyclopedia that is constantly moving. Wikipedia relies on collective knowledge in order to make its content evolve.

2.1 Participative but not Anarchic

Wikipedia allows the user to correct or increase the quality of any article. The user doing such a modification is called a "contributor". Every contributor is identified either by its account, if he created one, or his IP address at the moment of the publication.

Wikipedia encourages modifications, even minimal ones, because they can bring attention to an article thus making other users contribute on this article. This has two main consequences: firstly, it allows anyone to improve or damage articles[6]. Secondly, it also allows anyone to correct other's mistakes and damages. Wikipedia relies on that collective behaviour in order to evolve.

Unlike most websites or forums, on Wikipedia all users are equal. There are some administrators that are elected by users. They have some privileges that allow them to do some maintenance tasks and block users, but they don't have "full power" over users. Thus, if an administrator is in conflict with another user on the article about darwinism for example, he cannot impose his point of view.

[5] `http://www.alexa.com/topsites`

[6] Nevertheless, some pages are protected, e.g. Barack Obama's page during the presidential campaign. Protected pages can only be modified by users that have been registered for days or months.

However, contributions must follow some explicit rules. Rules are defined by users, the same way articles are, and their application is insured by the collectivity. For example, there are rules about the conditions[7] that an article should meet to be on Wikipedia. The online encyclopedia is thus a website that is not anarchic, but self-governed by processes defined by the users themselves [2].

There is, for example, a process which aims to assign categories to every article. This provides a structure to the encyclopedia by placing articles inside one or multiple categories. The Tennis article is inside "racquet sport" category which is itself contained in a category named "sport".

Also, multiple roles emerge on Wikipedia, some users, like the Wikignomes who browse articles in order to add references, categories, and correct dead links, ... "Cabalists" from the Mediation Cabal try to find a compromise between users with conflicting point of view. In this article we will focus on the "Recent Changes Patroller" role, which consists in a surveillance of the recent changes on the encyclopedia to prevent damages.

2.2 The RC Patrol

The sizeable activity of users on Wikipedia play a preponderant role in its evolution. In 2009, there are about 20 modifications per minute on Wikipedia (in French)[8]. Among these modifications, there are many quality contributions, but also many damaging ones. Some people use the articles as a means of chatting by successive modifications of them. Some others just want to verify if it is really possible to modify the article by adding test messages. Finally, there are some users that just want to damage the encyclopedia, by adding insults, libelous content, or by changing the meaning of an article. Those users making voluntary damages are called "Vandals".

The RC Patrol is a group of users, with no specific power and self-proclaimed, aiming at protecting the encyclopedia from being damaged. Patrollers follow the evolution of the "recent changes" page. Their objective is to cancel as quick as possible any modification that damages article integrity. Such a cancellation is called a "revert". Reverts sometimes come with a small comment, if the patroller decides to provide one.

Taking into account the number of modifications per minute, it is impossible for the patrollers to verify in details every modification. When a vandal adds insults inside an article, there is no doubt about the goal of that modification. However, when a contributor modifies a scientific article, the verification could both take a long time and require some expertise.

Thus, the RC Patrol tries to reduce the number of modifications to check, by warning vandals and reverting their modification in order to discourage them from damaging the encyclopedia. A vandal that decides to keep on damaging the articles after having being warned will be blocked by an administrator upon the request of a patroller.

[7] `http://en.wikipedia.org/wiki/WP:NN`

[8] `http://toolserver.org/~gribeco/stats.php`

Besides vandalism, some users can add other mistakes, by lack of expertise for example. Even patrollers can make mistakes or may not agree. As a revert is also a modification, users can therefore cancel the erroneous revert.

Wikipedia is an environment in which users can act freely. This environment is open, meaning that anyone can participate. User's autonomy, system openness and decentralised control of Wikipedia allows to consider it as a multiagent system and to apply existing approaches for related problems in this domain, like trust towards other agents of the system.

3 The ForTrust Trust Model

In this paper, we aim at using a trust model in order to assist the RC patrollers. As Wikipedia is open and decentralized, multiagent systems are suitable for that application. In this part, we present trust in multiagent systems and the ForTrust model, a multiagent trust model based on Castelfranchi & Falcone Social Trust Theory.

3.1 Trust in Multiagent Systems

The open and decentralised context of multiagent systems is interesting in terms of flexibility, adaptability and scalability but it also brings some risks and vulnerability. Agents, potentially developed by different people and acting in an autonomous way, can freely enter or leave the system, contribute to collective activities or transmit data to other agents. Thus, they can have a selfish behaviour, meaning that they will favour their own goals without taking into account other's goals or system integrity. This kind of agent can harm the system if it doesn't know how to react against such an agent.

The problem raised by the possible presence of selfish agents leads to a trust management problem towards other agents. Grandison [3] defines trust management as "the activity of collecting, codifying, analysing and presenting evidence relating to competence, honesty, security or dependability with the purpose of making assessments and decisions regarding trust relationships for Internet applications". Such a decision must be used in addition to classical security techniques which insure authentication, confidentiality of information, ... but that cannot garantee the behavior of transaction partners.

In multiagent systems, and generally speaking in many of the open web applications, trust management is often handled by reputation mechanisms. Some of those mechanisms work in a centralised way, for example, by using recommandations and opinions of the website users (cf. eBay[9], Amazon[10], ...). These opinions can be presented as they are or interpreted by an aggregation function (e.g. Sporas [4]). Other mechanisms, inspired by the multiagent field, work in a decentralised manner (e.g. Repage [5]) by allowing each agent to evaluate locally its neighbour's reputation such that the agent decides whether to trust or not.

[9] `http://www.ebay.com/`
[10] `http://www.amazon.com/`

Nevertheless, reputation is a single element among others that can be used to make trust assessments. In this article, we focus more specifically on the trust *decision*.

3.2 Social Trust Theory

According to Castelfranchi & Falcone [1] (C & F), social trust relies on four elements: a truster i, a trustee j, an action α and i's goal φ. C & F propose a definition of trust based on four primitive concepts: capacity, intention, power and goal. They state that: "an agent i trusts an agent j for doing the action α in order to achieve φ" iff:

1. i has the *goal* φ;
2. i believes that j is *capable* of doing α;
3. i believes that j has the *power* to achieve φ by doing α;
4. i believes that j *intends* to do α.

3.3 Trust Formalisation

A formal model of trust, relative to the social trust theory by C & F has been proposed in a precedent work [6]. This model is briefly described here, details about the formalisation realised in multimodal logic (called $\mathcal{L}$) which combines dynamic logic and BDI, can be found in the article [6].

That formalisation points out the difference between *occurent* trust and *dispositional* trust. Occurent trust represents a trust decision "here and now". In this case, the truster i has goal φ and trusts j to do action α now to achieve goal φ. In this article, we will only use the occurent trust, defined this way:

$$\begin{array}{l} Trust(i,j,\alpha,\varphi) \stackrel{\texttt{def}}{=} \\ \quad Goal(i,\varphi) \wedge \\ \quad Bel(i, Act(j,\alpha)) \wedge \\ \quad Bel(i, Power(j,\alpha,\varphi)) \end{array} \tag{1}$$

That formalisation uses the four primitive concepts by C & F, the predicate $Act(j,\alpha)$ meaning that j does α. It covers both capacity and intention: $Act(j,\alpha)$ is the case when j has the capacity and the intention of doing α. Predicate $Power$ means that j has the power of achieving φ by doing action α.

3.4 Trust in Inaction

Trust, as considered here, allows to represent how an agent can make a trust assessment towards another agent trusting that he will *act* in a certain manner to achieve a given goal. Nevertheless, Lorini and Demolombe [7] says we also have to consider trust in *inaction*, relating to the trust assessment made when an agent i trusts an other agent j so that j does not execute action α that can prevent i from achieving φ. Trust in inaction is defined as follows:

$$
\begin{aligned}
&Trust(i, j, \sim \alpha, \varphi) \stackrel{\text{def}}{=} \\
&\quad Goal(i, \varphi) \wedge \\
&\quad Bel(i, \neg Act(j, \alpha)) \wedge \\
&\quad Bel(i, Power(j, \alpha, \neg\varphi))
\end{aligned}
\tag{2}
$$

Meaning that i trusts j not to do α when i has goal φ iff: i has goal φ, i believes that j has the power to prevent i from achieving φ by doing α, and i believes that j will not do α (he has no capacity or no intention).

4 Trust Decision on Wikipedia

This section defines an application of the the ForTrust model, presented in section 3, to Wikipedia. Section 4.1 identifies actors, actions and goals and the situations where a trust decision happens. The way those contributions are inferred is presented in sections 4.2 and 4.3, respectively for the case of contribution needing correction and vandalism to revert. Finally, section 4.4 presents an implementation of an assistant agent using this trust model.

4.1 Application of the ForTrust Model to Wikipedia

The ForTrust model is used on Wikipedia in order to decide whether to trust or not a contributor. Trustees (j agent in the previous section) are Wikipedia contributors. Two actions are taken into account: page *modification* and *vandalism*. Even though a vandalism is a modification, we decide to take into account these actions separately in order to simplify the formalisation. The goal that we consider here is the role of the RC patrol: maintaining pages integrity. The achievement status of this goal is estimated by the patrollers themselves who decide if a page content integrity is maintained or not.
These two actions lead to two situations where one must decide whether to trust or not a contributor. In the first case, "patroller i trusts j to modify page p while maintaining its integrity" can be formalised this way:

$$
\begin{aligned}
&Trust(i, j, modify(p), integrity(p)) \leftarrow \\
&\quad Goal(i, integrity(p)) \wedge \\
&\quad Bel(i, Act(j, modify(p))) \wedge \\
&\quad Bel(i, Power(j, modify(p), integrity(p)))
\end{aligned}
\tag{3}
$$

An implication is used here instead of an equality (as used in the previous section), in order to get closer to the inference mechanisms implemented in logic programming (cf. section 4.4).
The second case, "patroller i trusts contributor j *not* to vandalize page p so that the page integrity is maintained" can be defined this way:

$$
\begin{aligned}
&Trust(i, j, \sim vandalize(p), integrity(p)) \leftarrow \\
&\quad Goal(i, integrity(p)) \wedge \\
&\quad Bel(i, \neg Act(j, vandalize(p))) \wedge \\
&\quad Bel(i, Power(j, vandalize(p), \neg integrity(p)))
\end{aligned}
\tag{4}
$$

We assume that if $Trust$ cannot be inferred, it implies $\neg Trust$.

Actions and goals are now clearly identified, the next step is to develop mechanisms that allow the agent i to infer those trust assessments from past behaviours of agent j. More precisely, those mechanisms must allow i to evaluate if the predicates $Power(j, \alpha, \varphi)$ and $Act(j, \alpha)$ are true, so that i can deduce j's reliability.

4.2 Trust in Action: Modification

In this case, we will focus on the contributor's power when he publishes a modification. Intentional vandalism is not considered in this part. The trust model is used in order to evaluate if a contributor reaches the wanted goal ($integrity(p)$) when he publishes some contribution (action $modify(p)$). Contributor's capacity and intention ($Act(j, modify(p))$) are always true and the decision to trust relies on the agent belief on $Power(j, modify(p), integrity(p))$. This belief is inferred the following way:

$$\begin{array}{l} Bel(i, Power(j, modify(p), integrity(p)) \\ \leftarrow category(p, c) \wedge \\ \quad image_m(j, c) > \delta \ \wedge \ \#O^{j,c} > 0 \end{array} \tag{5}$$

where the predicate $category(p, c)$ links page p to its category c and $image_m(j, c)$ is a function that maps to each pair (agent,category) the "quality" of j contributions in that category ($image_m : AGT \times CAT \rightarrow [0, 1]$ where AGT is the set of all agents and CAT the set of all categories) and $\#O^{j,c}$ is the number of changes done by agent j in the category c. The δ threshold is used to define the minimal value of the required image in order to consider that an agent produces contributions of reasonnable quality in a given category. The image function is defined this way:

$$image_m(j, c) = \frac{min(\bar{x}^{j,c}, \eta)}{\eta} \tag{6}$$

$$\bar{x}^{j,c} = \frac{\sum_{\langle t_s, t_e \rangle \in O^{j,c}} t_e - t_s}{\#O^{j,c}}$$

where $O^{j,c}$ is the set of all changes done by agent j in pages of category c. Each member of this set is a tuple $\langle t_s, t_e \rangle$ where t_s is the time of the modification, and t_e is the time when this page is modified again by another contributor. Thus, the difference between t_e and t_s represents the duration while agent j modification was not changed. Mean persistance of modifications of j in category c is given by $\bar{x}^{j,c}$. η is an upper bound for $\bar{x}$ such that if η is equal to one month, agents with $\bar{x} = 1$ month or $\bar{x} = 10$ months will get the same image. Thus, the general image of Wikipedia contributors relies on the persistence of their modifications.

In some particular cases, this metric can be noisy and therefore, a good contributor can receive negative feedback. This will happen, for example, when the contributor modifies a specific part of an article and another part of the same article is modified the following minute by another contributor. The first contributor will be regarded as a contributor with a low persistence of his modifications. But the only information source that is available is the Wikipedia website, thus it is impossible to use other metrics like the number of views per version of an article to reduce this noise. We also do not want to go into complex or unfeasable computations like the semantical analysis of the modifications.

4.3 Trust in inaction: Vandalism

This time, we need to use trust in inaction. Patrollers' goal stays the same ($integrity(p)$). Nevertheless, trust assessment does not rely on the *power* of a contributor to prevent the truster from achieving this goal by doing action $vandalize(p)$ (we suppose that every contributor has that power), but on his *intention* to do it (predicate $Act(j, vandalize(p))$). Belief inference on Act is done as follows:

$$\begin{array}{l} Bel(i, \neg Act(j, vandalize(p)) \\ \leftarrow (1 - image_v(j)) > \epsilon \ \wedge \ nc(j) > 0 \end{array} \tag{7}$$

$$image_v(j) = \frac{pvt(j)}{nc(j)} \tag{8}$$

where $pvt : AGT \rightarrow \mathbb{N}$ is a function that maps each agent j to the number of modification that has been labelled as vandalism and $nc : AGT \rightarrow \mathbb{N}$ maps to the total number of modifications done by agent j. $image_v(j) = 0$ means that no change has ever been considered as vandalism. The threshold ϵ is used to define the minimal image required so that Act becomes true. Agents' image regarding vandalism acts does not refer to a specific category because it does not rely on a specific expertise in a category of articles.

4.4 From the Abstract Model to an Implementation

An implementation of an agent using the ForTrust model has been realized with a BDI architecture using the *Jason* language [8]. *Jason* fits well an implementation of a formal definition of trust, as defined in section 3: it is based on logic programming and BDI architecture at the same time.

Fig. 2 illustrates the main components of the agent architecture. Data structures encapsulates goals, beliefs, intentions and plan library. The *perceive* process updates the belief base from the environnement and the *act* process selects the next action to be done from the set of current intentions. Finally, the *inference* process must decide whether to trust or not an agent.

Specialisation of this agent to the Wikipedia specific context is easy while using *Jason*. Decisions to trust have to be customized and some inference rules

```
1  // inference rules for Trust decision, Goal=integrity(p)
2  trust(J,modify(p),Goal)[strength(C)] :-
3     .intend(Goal) &                          // the agent has the goal
4     act(J,modify(p))[strength(X)] &          // J has is capable and intends
5     power(J,modify(p),Goal)[strength(Y)] & // J has the power
6     C = math.min(X,Y).                       // computes trust strength
7     // trust strength is represented by annotations in hooks
8
9  trust(J,~vandalize(P),Goal)[strength(C)] :-
10    .intend(Goal) &                                  // the agent has the goal
11    act(J,~vandalize(P))[strength(X)] &
   // J is capable and intends to not vandalize
12    power(J,vandalize(P),~Goal)[strength(Y)] & // J has the power
13    C = math.min(X,Y).
   // computes trust strength
14
15 // P integrity is a goal to maintain
16 { begin mg(integrity(P)) }
17 { end }
18
19 //rules for action (capacity and intention)
20 act(J,modify(p))[strength(1)].
21 act(J,~vandalize(P))[strength(X)] :-
22    (1-imagev(J)) > 0.8.
23
24 // rules for power
25 power(J, modify(p), _)[strength(Y)] :-
26    category(P,C) & imagem(J, C) > 0.5.
27    // the image function is implemented so that the 3rd term is
28    // the value for agent J in category C
29 power(J, vandalize(P), _)[strength(1)].
30
31 // computing images:
32 imagev(J,0) :- 0 = .count(modify(_,J,_)).
33 imagev(J,X) :- X = .count(vandalize(_,J,_,_)) / .count(modify(_,J,_)).
```

Fig. 1. Extract of *Jason* implementation

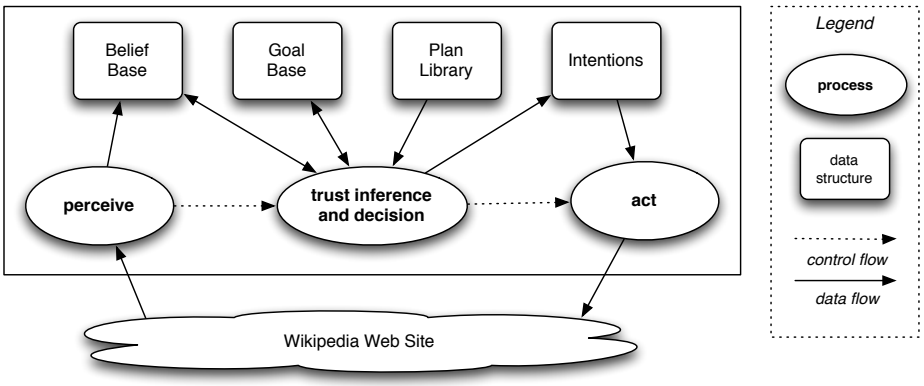

Fig. 2. General architecture for trust

must be added for that decisions. It is also needed to implement the perception component which analyses the "recent changes" page on Wikipedia and translates it into perceived facts.

The perception given by the architecture is then converted into first order predicates and included in the belief base whith a specific annotation indicating that they correspond to the agent perception. Older perceptions are removed consequently. Among those beliefs, the following are extracted from Wikipedia:

- $modify(p, j, t)$: agent j modified p at time t.
- $vandalize(p, j, g, t)$: change on page p by j at time t was labelled as vandalism by g.
- $category(p, c)$: page p belongs to category c.

An intitial implementation of this agent in *Jason* is presented in Fig. 1. Lines from 1 to 13 implement the trust inference mechanisms. Lines from 20 to 29 implement the inference of predicates *Act* and *Power* as defined in (7) and (5). Lines 32 to 33 implement the $image_v(j)$ function computation (8). The $image_m(j, c)$ (6) is computed in a similar manner but is not reproduced in this source code extract for simplicity reasons.

5 Trust-based assistance to Wikipedia patrollers

The encyclopedia quality relies directly on each user's contribution and correction to existing articles. Most important damages come from vandalism acts that consist in erasing valid content or introducing voluntarily false data inside articles or text without relation to the article topic. The case of imperfect contributions (incomplete content, wrong spelling, ...) also has an impact on the global quality. We propose to use trust in order to spot contributors that need to be frequently corrected. Trust assessment towards contributors will allow to spot untrustworthy users and target verifications towards them.

A complete automation of the decision to revert a contribution, to correct it or to block a user will be really difficult to realize. The verification task is currently handled by human patrollers because it requires a semantic interpretation of articles content, which is currently impossible to realise with a software. There are some robots which do very simple verifications, by seeking terms or characteristic regular expressions, in order to prevent the insertion of insults in articles. But they are not reliable and show a large number of false positives. Therefore, the patroller has to judge whether a contribution is a vandalism act when he thinks that the damage was intentional. This decision is subjective, and the patroller can make mistakes. For example, what if a contributor indicates a wrong date for a historical event, is it an error or voluntary damaging?

Over the 30000 daily modifications on wikipedia (French), the patrollers' task is a hard one. Fig. 3 shows the number of overall daily deletions and reverts[11]. Reverts represents only 3 to 6% of overall contributions. But it is necessary to control every contribution in order to identify the ones that are vandalism acts. Fig. 4 estimates the revert delay of a vandalism by indicating the probability that it can be reverted in a certain amount of time[11]. If approximatively 50% contributions are reverted in less than 2 minutes, a non negligible quantity (approximatively 25%) stays online longer than 1 hour and about 10% remains longer than one day. Most likely, the cause of this delay is the huge number of contributions to check.

Decision	Assistant Agent's beliefs
VAN	$\neg \mathit{Trust}(i, j, \sim \mathit{vandalize}(p), \mathit{integrity}(p)) \ \wedge \ \mathit{nc}(j) > 0$
COR	$\mathit{Trust}(i, j, \sim \mathit{vandalize}(p), \mathit{integrity}(p)) \wedge$ $\neg \mathit{Trust}(i, j, \mathit{modify}(p), \mathit{integrity}(p))$
INT	$\mathit{Trust}(i, j, \sim \mathit{vandalize}(p), \mathit{integrity}(p)) \wedge$ $\mathit{Trust}(i, j, \mathit{modify}(p), \mathit{integrity}(p))$
UKN	$\mathit{nc}(j) = 0$

Table 1. Relation between beliefs and decisions (i is the patroller, j is the contributor and p the modified page).

We propose to use the ForTrust model, as described in section 4, in order to assist patrollers in their verification tasks. Our goal is to reduce their task charge by supplying them with an assistant agent which has the ability to make trust assessment in order to help the patroller targetting his verifications.

The assistant agent can therefore use its own trust model in order to classify contributions in four distinct categories:

– **VAN**: contributions that probably are vandalism;

[11] Data and graphics from `http://toolserver.org/~gribeco/stats-vandalism.php`, based on `http://fr.wikipedia.org/` project.

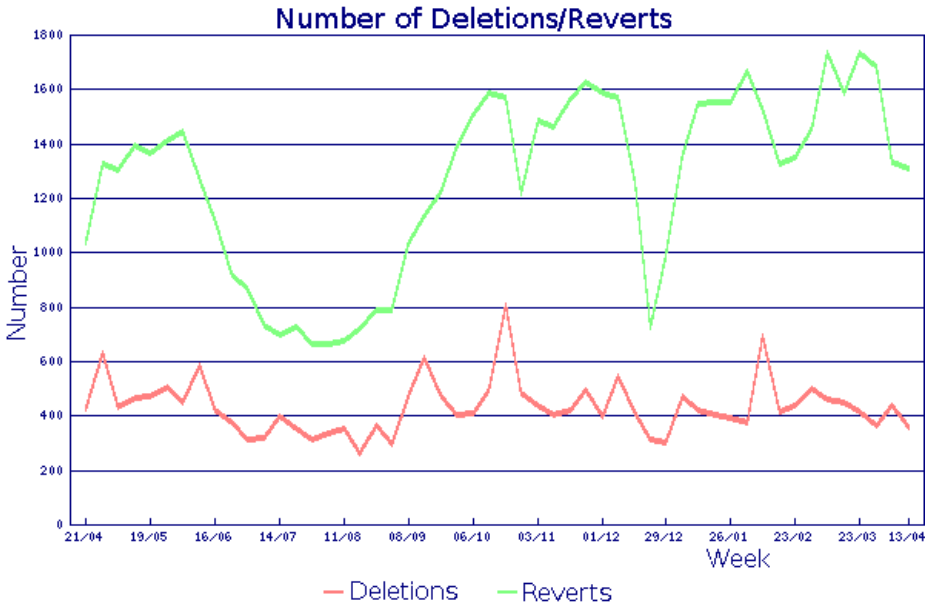

Fig. 3. Deletion and reverts on Wikipedia (retrieved on april 22nd, 2009)

- **COR**: contributions that may need to be corrected;
- **INT**: contributions with less chance of being reverted or corrected;
- **UKN**: the agent is not sure about these contributions.

The classification of a contribution in one or another category depends on the trust assessment that the agents makes towards the contributor. Table 1 summarizes the relationship between decision and agent beliefs.

This table illustrates the advantages of a cognitive model for an assistant agent for the patrollers. The use of C & F socio-cognitive theory allows to describe to a human user the reasons that brings the agent to a trust or distrust decision. Here, we aim to distinguish between distrust regarding an intentional vandalism and distrust regarding a low skill contribution. The assistant agent does not replace the patroller but gives advice by telling him that some contributions do not need to be checked (those in category **INT**). It could also completely mask the **INT** contributions in order to reduce significantly the amount of contributions to verify.

It is interesting to note that the trust model of an assistant agent is constantly evolving and can review its trust assessment. For example, if a contributor stops being a vandal and decides to contribute correctly, the trust model will change its trust value accordingly after a certain time. On the other hand, trusted contributors may not be verified directly by patrollers and a trusted contributor that has turned vandal may not be spotted immediately. Eventually, every contribution is checked, as the "classical" means of article surveillance still are available:

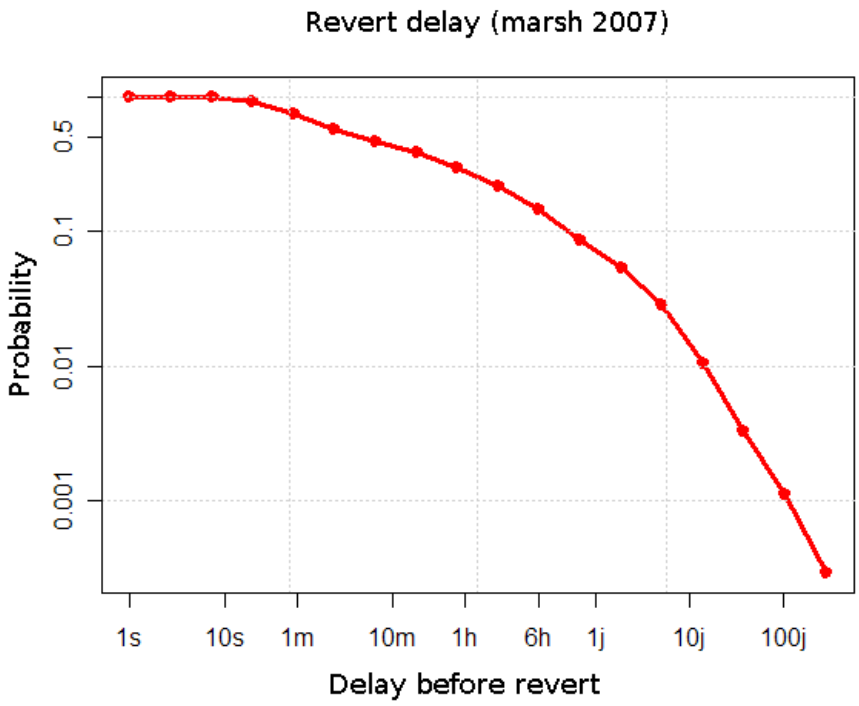

Fig. 4. Probable delay before revert

- The Watchlist is a list managed by the user that displays all changes that have been done on the articles selected by the user.
- The Projects: Some users participate in projects and regularly checks articles contained in the project to ensure their correctness.
- Article Browsing: Mistakes and vandalisms can be detected by users that are simply gathering information on Wikipedia.

Once the vandal has been spotted, the assistant agent will adjust its trust value and the contributor could be marked as untrusted.

6 Conclusion

In this article, we presented an application allowing Wikipedia patrollers to automatically filter recent changes in order to verify only those performed by untrustworthy contributors. This is obtained by providing to the patrollers an assistant agent that can make trust assessments based on the contributor's supposed capacities, intentions and power. The use of a cognitive trust model allows

to reason on the distrust decision to distinguish between those who are vandals and those who are low skill contributors.

The implementation of a BDI agent is proposed here with the *Jason* language. It relies on the website's changes, freely available, to perceive contributions. The agent watches patrollers' actions, namely modification or revert, and learns the trust assessment that is made towards contributors. This trust model is then used to reduce the amount of modifications to verify based on trust assessment toward their contributors.

A simulation of these assistant agents is currently being implemented in order to validate experimentally the pertinence of the decision taken by the agents: By using the contribution's history (including reverts) available on Wikipedia, we will then simulate a learning and an assistance to patrollers. The distance between assistant assessment and real decisions (from the patrollers) will then be correlated in order to evaluate a distance between both. Model parameters, like decision thresholds, can be decided experimentally by doing multiple simulations in order to reduce this distance.

A large number of wikis work the same way, thus, even if the application described here relies on Wikipedia, it is possible to transpose it on other wikis, e.g. those of the Wikimedia foundation (Wiktionary, Wikiversity, ...).

Acknowledgements

The work presented in this article is supported by the ANR in the ForTrust ANR-06-SETI-006[12] project.

References

1. Castelfranchi, C., Falcone, R.: Social trust: A cognitive approach. In Castelfranchi, C., Tan, Y.H., eds.: Trust and Deception in Virtual Societies. Kluwer (2001) 55–90
2. Viégas, F., Wattenberg, M., McKeon, M.: The hidden order of Wikipedia. Lecture Notes in Computer Science **4564** (2007) 445
3. Grandison, T., Sloman, M.: Trust management tools for internet applications. In Nixon, P., Terzis, S., eds.: First International Conference on Trust Management (iTrust). Volume 2692 of LNCS., Springer (May 2003) 91–107
4. Zacharia, G., Moukas, A., Maes, P.: Collaborative reputation mechanisms in electronic marketplaces. In: Proceedings of the Hawaii International Conference on System Sciences (HICSS-32). Volume 08., Maui, Hawaii, United States of America, IEEE Computer Society, Washington, DC, United States of America (1999) 8026
5. Sabater-Mir, J., Paolucci, M., Conte, R.: Repage: Reputation and image among limited autonomous partners. Journal of Artificial Societies and Social Simulation **9**(2) (2006) 3
6. Lorini, E., Herzig, A., Hübner, J.F., Vercouter, L.: A logic of trust and reputation. Logic Journal of the IGPL (2009)

[12] `http://www.irit.fr/ForTrust/`

7. Lorini, E., Demolombe, R.: Trust and norms in the context of computer security. In Springer-Verlag, ed.: Proceedings of the Ninth International Conference on Deontic Logic in Computer Science (DEON'08). Volume 5076 of LNCS. (2008) 50–64
8. Bordini, R.H., Hübner, J.F., Wooldrige, M.: Programming Multi-Agent Systems in AgentSpeak using *Jason*. Wiley Series in Agent Technology. John Wiley & Sons (2007)