WiseShare: A Collaborative Environment for Knowledge Sharing Governed by ABAC Policies

Hakima Ould-Slimane Moustapha Bande Hanifa Boucheneb

Laboratoire VeriForm, Department of Computer Engineering, École Polytechnique de Montréal, Quebec, (Qc), Canada.
hakima.ould-slimane@polymtl.ca, Moustapha.bande@polymtl.ca, hanifa.boucheneb@polymtl.ca

***Abstract*—In this paper, we propose an attribute based access control (ABAC) approach for safely sharing knowledge in a collaborative environment. Indeed, existing similar systems facilitate collaboration at the risk to convey doubtful information and sometimes serve as a gate to vandalism. Our system called "WiseShare" ensures collaboration while focusing on the reliability of the broadcasted content. To achieve this goal, we precisely express the requirements needed to control the shared information. In addition, we define a formal framework for specifying security policies governing contributions requests according to user profile. We present also a prototype implementing the functionalities of our system. In our approach, a new user has limited rights. On an ongoing basis, if he demonstrates his ability to produce good contributions, he gains new privileges. Conversely, if he often generates contributions with a questionable content, he loses the held privileges. Therefore, granting user requests depends closely on his previous behavior (the history). Indeed, our system encourages users to adopt a responsible attitude by focusing on the reliability of the content of their contributions instead of their frequency. Consequently, WiseShare permits to significantly minimize clumsy and malicious actions.**

***Index Terms*—Collaborative environments, knowledge sharing, attribute based access control.**

I. Introduction

During the last decades, the need of sharing information induces the large proliferation of collaborative environments. Indeed, using this computing fashion results in substantial gain both in the output quality and in the reduction of calculation time. These collaborative systems are widely and successfully used by shared computation systems and collaborative edition platforms. Moreover, with the emergence of systems based Web 2.0, the collaboration concept becomes essential for the viability of many popular applications such as Wikipedia, Flickr, Youtube, Del.icio.us, etc. However, the large sharing and manipulation of information between users in a collaborative environment impose a tight monitoring on their activities. Usually, to ensure trustworthiness of the shared resources, access control policies must be defined and enforced during the collaboration. In the literature, there are many access control models [2],[3],[4],[8] each of them focuses on a particular dimension related to users, resources or relation between them. Generally, users are seeking for flexible systems that allow them to easily share information. However, ensuring flexibility and efficiency of the collaborative systems while controlling the access to the involved resources is a great challenge.

Although, access control models in collaborative systems are various and heterogeneous, there are some common and specific requirements that can be defined for their evaluation. In [1], the authors present basic requirements that any access control model in a collaborative environment should meet. Thus, an effective access control model:

- should be generic and expressive;
- requires great scalability;
- must be able to protect information and resources of any type and at varying level granularity;
- must facilitate transparent information access to authorized users and strong exclusion of unauthorized ones in a flexible manner;
- must allow high-level specification of access rights;
- must be dynamic (specify and change policies at runtime);
- should keep performance and resource costs within acceptable bounds

Driven by the public passion for online information search, we investigated the famous collaborative encyclopedia "Wikipedia". After a critical review, we could enjoy the usability and the richness of the encyclopedia. However, from a security point of view concerning the management of the contribution content, we identified some drawbacks mostly due to the high accessibility that the platform offers to users. In fact, this popularization enables an immediate dissemination of each contribution independently of its content, and thus, is conducive to inappropriate and malicious activities of users. To overcome this hyper-permissiveness, we propose in this paper a collaborative system that we called "WiseShare" for

COLLABORATECOM 2012, October 14-17, Pittsburgh, United States

DOI 10.4108/icst.collaboratecom.2012.250402

creating and sharing knowledge while security is being closely ensured. More precisely, we aim to control the collaborators activities so that only reliable contributions are broadcasted. For our design purposes, we need an access control model that is capable to define fine-grained level permissions and offers the possibility to handle dynamic changes on our collaborative environment. Consequently, in our system, users may have different permissions for their contributions. These permissions are dynamically and automatically generated by the system which is based on the profile and the behavior of users through their contributions. The system may update the profile of a user each time he submits a contribution. It is clear that the definition of static roles cannot deal with our system requirements. Thus, the ideal candidate for our specific design purposes, is the ABAC model (Attribute Based Access Control) [8]. Indeed, ABAC proposes the definition of permissions based on any security-relevant characteristics (the attributes). This model is suitable for our approach since we are planning to control the collaboration system through the contributor's attributes, the contribution attributes, as well as the context of the collaboration. The rest of this paper is organized as follows. Section II discusses the related work concerning access control models for collaborative systems. Section III focuses on social based collaborative environments and provides a critical overview of "Wikipedia". Section IV describes the foundations of our approach, the requirements and the targeted objectives. Section V provides a formal framework for specifying the components of our approach (user, contribution and context) as well as the ABAC policies reflecting our requirements. Section VI presents the algorithms depicting the impact of performing contributions on users profile according to the defined ABAC policies. Section VII depicts a prototype implementing our system (WiseShare). Finally, Section VIII provides some concluding remarks and future work.

II. Related Work

In the literature, several access control models have been proposed for collaborative systems. Role-based access control (RBAC) which has been proposed by Sandhu et al. [2] is based on the organizational structure of the enterprise and maps permissions to roles and roles to users. Even though it is flexible and easy to manage, RBAC is not suitable for all collaborative systems since it cannot handle fine-grained control on individual users and doesn't include the impact of context in taking access control decision. To take into consideration contextual information, Task-based access control model (TBAC) [3] has been introduced as an extension of RBAC. TBAC introduces dynamic management of permissions during the progress of the task until its completion. While considering collaborative system access control, TBAC fails to be more efficient since it has to manage activation and deactivation of permissions during task execution. Team-based access control (TMAC) [4] and Contextual TMAC (C-TMAC) [5] are other extensions of RBAC and offer the possibility to specify fine-grained access control. They define context on user level and context on object level. However, TMAC and C-TMAC lack to clearly define the assignment of relations between entities. Team and Task-based RBAC (TT-RBAC) [6] adds the concepts of team and task into RBAC model while ensuring context-aware access control. In this model, users are assigned to teams, tasks and roles to permissions and permissions to tasks. Context-based access control (CBAC) [7] is another extension of RBAC by considering environment roles. In this model, roles are activated based on environment conditions at the time of the request. However, all these proposed models have some shortcomings while considering collaborative environments since they are either incomplete or addressed a specific need and subject to insufficient insurance. Instead of only considering subjects, objects and contexts, attribute-based access control (ABAC) has been proposed in[8]; it introduces the concept of attribute related to the different entities involved in collaboration to make the access control decision. Smari et al. [9] proposed a scheme to incorporate trust and privacy in the ABAC model. In this scheme, the level of trust is affected by context and subject attributes and has an influence on the access control decision. Similarly, privacy preserving is also measured by ensuring that a requested object is used accordingly to the access purpose of the subject. To allow a large number of users to gain benefit of sharing, Wang et al. [10] proposed a novel ABAC framework based on people tagging. The main idea is to identify user attributes from the information provided by the collaborative efforts of the system users. The authors also propose a formal language for tag-based policy specification. This model may be vulnerable to malicious users' collision since one can gain faked tags from partner users. Nasirifard et al. [11] proposed an annotation-based access control model for collaborative and social platforms. The model also uses the concept of "tagging" for annotating shared resources and is applicable in multiple Web-based collaboration systems. However it lacks to take into account contextual information. Demchenko et al. [12] proposed a model using ABAC for securing web services and Services Oriented Architectures (SOA). They discuss also a detailed comparison between RBAC and ABAC models.

III. Social Collaborative Environments

The social computing (web 2.0) implies any form of technological progress that makes the web more simpler, more accessible and mainly more ergonomic. This innovative environment allows users from different horizons to take full advantage of the web functionalities as well as the online conveyed content. Moreover, this revolution turns the common passive user checking only information into an active actor. Hence, the users have voluntary responsibilities in content creation and sharing. In fact,

large web collaboration allows popular application blossoming such as Wikipedia (the collaborative encyclopedia) and Del.icio.us (bookmarks sharing tool). Users have the choice and the power to collaborate by bringing a contribution, as little it can be. Thus, every user of the collaborative system has the right to benefit from the whole information even though he did not contribute. Faithful to the web 2.0 philosophy, these systems offer a large flexibility for broadcasting any contribution (regardless of its source) which may be subject to the continuous updates of users. In fact, the control is ensured in a collaborative and voluntary way. In other words, the main purpose of these systems is to encourage sharing and collaboration instead of their restriction. From a security point of view, it represents a big potential of hidden vulnerabilities in this hyper-permissive collaboration. In order to better illustrate our opinion, we have investigated the free encyclopedia Wikipedia. Actually, this online encyclopedia is one of the most visited search engines on the web. Most people are using it as the first, even the single reliable reference. The information broadcasted by this encyclopedia is collected in a collaborative way. Moreover, Wikipedia's catchword is: "everyone is welcome to contribute". Fortunately, Wikipedia is designed mostly by the collaborative efforts of honest contributors who share voluntarily their knowledge. However, it would be careless to give absolute confidence to all the content disseminated in Wikipedia. Indeed, it is impossible to ensure either the honesty or the competence of all the contributors being active at any time on the system. At the organizational level, Wikipedia's volunteers are classified by their status. Therefore, the underlying access control model is the RBAC model. It mostly contains the following roles: administrators, bureaucrats, IP address inspectors and the bots (spell-checkers). In Wikipedia, any registered user has the right and the privilege to immediately disseminate his contribution regardless of its content and without any verification. Therefore, a contribution with an inaccurate or malicious content may be visible until someone intervenes to modify or remove this contribution. Moreover, the expertise notion does not exist for the contributors; there are only thematic portals for classifying the contributions. Any user can also modify a contribution posted by another one, mostly even though he is not skilled to do it. He can also disseminate his modifications without being authenticated. In this case, the IP address of such a user is publicly revealed. To prevent inappropriate modifications or vandalism actions, a history of all the modifications is kept. This history is used to undo an inadequate modification in order to get back to a previous correct version. Nevertheless, it is sometimes difficult to retrace the history to reach the original version. In addition, the prevention of inappropriate and malicious actions is ensured by a category of selected people (administrators and bureaucrats). So, their presence in the encyclopedia system is required all the time since they are also responsible for blocking users (who behave as vandals) as well as promoting them to privileged rank. Therefore, every administration intervention is done manually based only on the volunteers' decision. Consequently, the collaboration is widely facilitated compared to the reliability of the disseminated information. This is the issue that led us to think about balancing the goal of flexible collaboration and reliable contribution. Mainly, we aim to propose an automatic mechanism for accessing and controlling user's profile trough their activities (contributions). These profiles are implemented according to the ABAC model. Furthermore, each contribution request must respond to a specific security policy and may potentially update its owner profile.

IV. WiseShare: Description of Our Approach

In this section, we describe "WiseShare" our ABAC based collaborative system for knowledge sharing. The main objective of this system is to allow a reliable sharing of knowledge where security is strongly enforced. Our system consists of three main components: the contributors (collaborators, users), the contributions (the knowledge being shared) and the environment (the context) under which the collaborative contributions occur.

The overall philosophy of our system is based on the following features:

- Do not give all users the same rights of contribution: Indeed, it would be wise to grant the contribution rights according to the user profile. So, an expert contributor may not have the same rights as a novice one.
- The access rights based on ABAC model are automatically generated from the contributor's profile. In our system, access control policies are mainly defined according to the profile characteristics of the contributor.
- The contributors are involved as little as possible in the management of the contributions content made by other users (editing tasks are limited to experts). That helps to preserve the enthusiasm of the contributors since they do not have to permanently monitor the new contributions or track the history of contributions.
- The contributor must register. Hence, there is no need to reveal their IP addresses because there is more control on background. Indeed, a safe collaboration requires an authentication to map users to their contributions.
- The global collaborative behaviors of contributors "decide" of their "rank" in the system. Indeed, unlike the work of Wang et al. [10], our approach is using a behavior-based tagging which is not based on users'opinion.

- Each request can potentially change the attributes of the system components (which are: users, contributions, environment). So, contributing becomes an activity taking place in a dynamic environment. It evolves according to the global contributions effect.
- Confidence is a volatile resource (no one is an expert forever) and is guaranteed only if the user behavior confirms it. Hence, there is a continuous assessment of the contributor behavior upon each issued contribution.
- We can distinguish three kinds of contributors: novice, expert and vandal. Each new user is noted as novice. If he demonstrates his expertise in a certain topic, he becomes an expert in this topic. However, if he is massively tagged as a malicious contributor, he turns into " vandal " and thus, he is blocked (he loses the right to contribute).
- Each new contributor has to demonstrate his expertise. The user builds his reputation according to his interventions, so he does not have to claim to be an expert. The expertise is proved if the user succeeds to disseminate a certain number of significant contributions in a specific domain.
- Each query (creating, posting, editing, suppressing, reporting vandalism, etc.) must satisfy an ABAC policy based on the contributor's reputation (automatically generated tag).
- When there is detection of suspicious behavior (from a novice or a vandal user), the experts are automatically notified to intervene.
- The ABAC Policies governing the system can include:
  - The query binding a user to a contribution, like: creating, editing (modification), posting, suppression.
  - Or the opinion a user can have about another user, like: reporting a vandalism case.

Mainly, the collaborative strategy adopted by WiseShare aims to achieve the following objectives:

- Controlling the contribution activities (and therefore sharing) to prevent vandalism and clumsy contributions.
- Reducing users intervention.
- Automatically profiling contributors according to their global contributory behavior.
- Optimal and automatic management of the history of contributions modifications (ideally suppressing it).
- Speed and automatic prevention and detection of vandalism.
- Encouraging honest contributors to step back by reviewing their contributions (pass it through a spell checker, check their sources...etc.).
- Automating as possible the processing of contributions.
- Updating the status of contributors according to the quality and the frequency of their contributions.

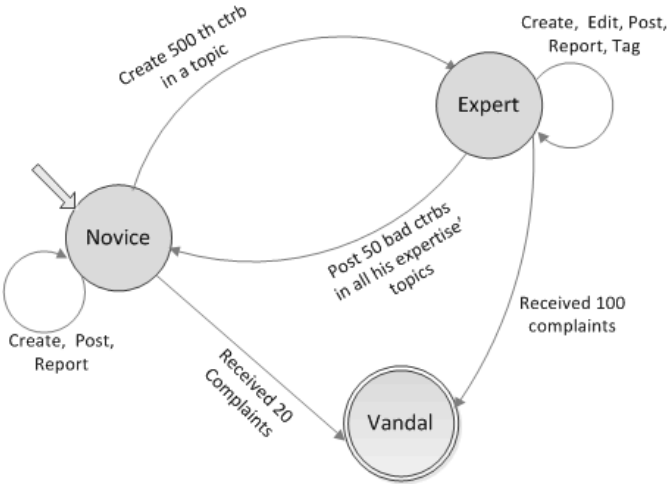

Fig. 1. Evolution of users'reputation in WiseShare.

A. Access Control Requirements In WiseShare

In this section, we present the outlines of our solution. We suppose that a contributor u would like to post a contribution c in a collaborative environment e controlled by a set of ABAC policies P. The figure 1 shows the evolution of the user's profile through the three main states (novice, expert and vandal).

Each contribution request r expressed by a user u has to satisfy the following principles:

- Every new user of the system is tagged "novice" by default.
- A novice user has just the right to create temporary pages which are only visible to the experts in the contribution topic. Hence, a novice contributor is unable to broadcast immediately his contributions to the public.
- The creation of temporary pages automatically notifies the experts for editing the new contribution if it is necessary.
- After one week, if the temporary contribution is neither suppressed nor radically changed, this contribution becomes visible for the public, and is counted (recorded) for the original author (in the concerned topic).
- If a novice achieves n (for instance 500) recorded contributions in a certain topic, he will automatically receive the tag "expert" in this topic.
- if an expert modifies drastically the content of a contribution, he becomes the main author of this contribution. This does not mean that the expert becomes the owner of the contribution idea. Indeed, the original author of the contribution never changes.
- If an expert contributor posts a m (for instance 50) irrelevant contributions (removed or significantly corrected contributions) in his field of expertise, he loses

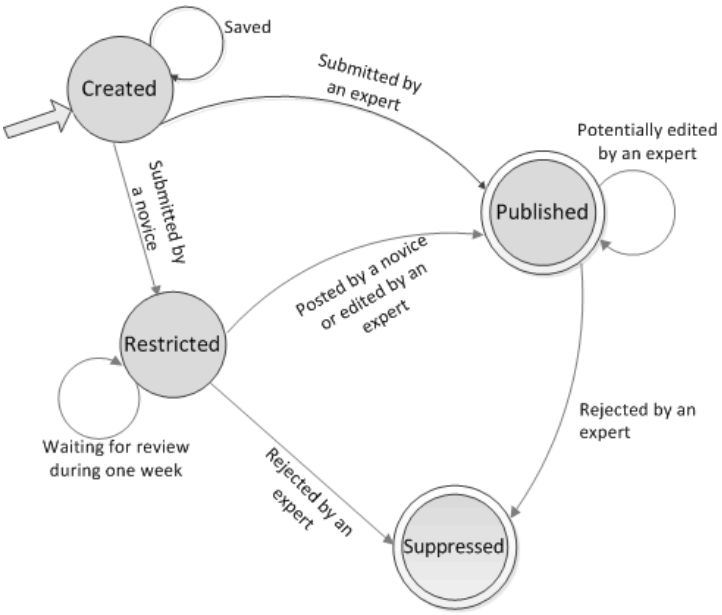

Fig. 2. The life cycle of a contribution in WiseShare.

the reputation of expert in this topic. If he is no longer expert in all the topics, he becomes unreliable and receives the tag "novice" and loses all his privileges.

- Every contributor can tag another contributor as "vandal" but he cannot denounce it more than once. In addition, the reporting of a vandalism case has to be justified.
- If a contributor is tagged as "Vandal" by k contributors (for instance 20 for a novice user and 100 for an expert user), he is automatically banned from the system. However, malicious users may collaboratively tag one honest user as Vandal and thus the system may block the correct user. To avoid this situation, we consider only complaints expressed by trustable users (for instance, trust can be measured by the user seniority within the system).
- A contributor user u can only express contribution requests r among the set {Create, Post, Edit, suppress, Report }.
- If the request r satisfies the corresponding policy from the set P, it is granted; otherwise, it is denied.

V. ABAC Policies Specification

In this section, we present our formal framework for specifying the attribute based access control policies required by WiseShare. We recall that our system consists of three entities: User (contributor), Contribution, Environment. Formally, each component is a tuple of attributes as follows:

- The set of user attributes $\mathcal{U}$: define the identity and the characteristics of a user who contributes in the system. A contributor u is represented by a tuple as follows: $u =< idu, rep, Skl, cb, cpl, Vdl >$ where:
  - ▷ idu is a unique string denoting the identity of the user who must register to the system.
  - ▷ rep denotes the reputation of the user, where $rep \in \{"novice", "expert", "vandal"\}$.
  - ▷ Skl denotes the set of expertise a contributor has gained during his collaboration in the system (the user skills, for instance: $Skl \subset \{"Art", "Sciences", "Sport", "Philosophy", ...\}$). By default, it is empty when he registers.
  - ▷ cb gives for each topic the number of recorded contributions proposed by this user. By assuming k topics, we have: $cb = \{cb.t_1, \ldots, cb.t_k\}$, where t_i is the i^{th} topic.
  - ▷ cpl denotes the number of the received complaints made by distinct users.
  - ▷ Vdl denotes the set of users identifiers denounced by this user.
- The set of contribution attributes $\mathcal{C}$: defines the features outlining a contribution. These attributes serve to locate and identify a contribution in the system, so that it cannot be confused with another one. A contribution c is given by a tuple: $c =< idc, d_{cr}, orig, chf, th, vis, ctt >$ where:
  - ▷ idc denotes the identifier of the contribution (a string).
  - ▷ d_{cr} denotes the creation date of the contribution in the system.
  - ▷ $orig$ denotes the original author of this contribution (he can be a novice or an expert).
  - ▷ chf denotes the main author of this contribution (he can be a novice or an expert).
  - ▷ th denotes the specific topic related to the contribution (we suppose that we have k topics sorted by alphabetical order), where $th \in \{t_1, \ldots, t_k\}$.
  - ▷ vis denotes the degree of the contribution visibility. Indeed, $vis \in \{created, restricted, published, suppressed\}$. In the case of a contribution created by a novice user, it has a restricted visibility, so it can be only viewed by experts (skilled in the contribution topic). When a contribution is approved by the experts or proposed by one of them, it becomes visible to everyone. When a contribution is suppressed by an expert, it becomes non viewable for all users. The life cycle of a contribution in WiseShare is illustrated in figure 2.
  - ▷ ctt denotes the content of the contribution.
- The set of environment attributes $\mathcal{E}$: describes critical information related to the collaborative context in which contributions occur. A collaborative environment e is given by the following tuple: $e =< date, Exp, BL >$ where:
  - ▷ $date$ denotes the current date of the system.

$$
\begin{array}{rcl}
\mathcal{P}_{Create}(u,c,e) & \Leftrightarrow & u.idu \notin e.BL \\
\mathcal{P}_{Post}(u,c,e) & \Leftrightarrow & u.idu \notin e.BL \wedge (u.rep = \text{"}expert\text{"} \\
 & \vee & \vee (u.rep = \text{"}novice\text{"} \wedge e.date - c.d_{cr} \geq 7)) \\
\mathcal{P}_{Edit}(u,c) & \Leftrightarrow & u.rep = \text{"}expert\text{"} \wedge c.th \in u.Skl \\
\mathcal{P}_{Supress}(u,c) & \Leftrightarrow & u.rep = \text{"}expert\text{"} \wedge c.th \in u.Skl \\
\mathcal{P}_{Report}(u_1,u_2) & \Leftrightarrow & u_1.idu \notin e.BL \wedge u_2.idu \notin u_1.Vdl
\end{array}
$$

TABLE I
ABAC POLICY SPECIFICATION IN WISESHARE

- ▷ Exp denotes the list of recognized experts sorted by topic.
- ▷ BL denotes the black list which contains only the signature of banned users (ex: the IP addresses of the vandals).

$\mathcal{A}(u)$, $\mathcal{A}(c)$, and $\mathcal{A}(e)$ are attribute assignment relations for user u, contribution c, and environment e, respectively:

- ▷ $\mathcal{A}(u) \subseteq idu \times rep \times Skl \times cb \times cpl \times Vdl$;
- ▷ $\mathcal{A}(c) \subseteq idc \times d_{cr} \times orig \times chf \times th \times vis \times ctt$;
- ▷ $\mathcal{A}(e) \subseteq date \times Exp \times BL$

In our system, an attribute based access control rule (policy) is given as the following predicate:
$\mathcal{P} : \mathcal{U} \times \mathcal{C} \times \mathcal{E} \longrightarrow \{true, false\}$
$\mathcal{P}(\mathcal{A}(u), \mathcal{A}(c), \mathcal{A}(e)) =$

$$
\begin{cases}
true & \text{if } u \text{ handles } c \text{ in } e \text{ according to } \mathcal{P} \\
false & \text{otherwise}
\end{cases}
$$

Indeed, given the attribute assignments, the evaluation of an ABAC based policy is reduced to the evaluation of first order logic expressions. By adopting the formalism defined above, we can now express all the requirements of WiseShare according to the ABAC model. The policies controlling the different contributions a user can perform are presented in TableI. For instance, the second rule $\mathcal{P}_{Post}(u,c,e)$ states that: a user u can post a contribution c in a context of collaboration e, if and only if, he doesn't belong to the black list (he is not a vandal), and he is an expert user or a novice user who is the original author of a contribution submitted since more than a week.

VI. WiseShare Security Enforcement

In this section, we will present the different pseudo-algorithms depicting how a contribution request can affect the different attributes of the system components, especially the contributor attributes. A user in WiseShare can perform one of the following actions: Create, Post, Edit, Suppress and Report. However, before performing such interventions, he must submit a contribution request. If the request is accepted (i.e., satisfies the corresponding policy), he can actually perform his contribution. Otherwise, his contribution will be suspended until the corresponding policy requirements are fulfilled at some moment in the near future. The pseudo-algorithms enforcing WiseShare policies for the different queries are presented in: figure 3, figure 4, figure 5, figure 6 and figure 7. Based on the system attributes values, each algorithm decides whether the required action will take place or not. In case the required action is executed, the algorithm performs updates on the system attributes accordingly.

VII. WiseShare Prototype and Functionalities

In this section, we present the prototype implementing our knowledge sharing collaborative system WiseShare. We have developed this prototype using PHP language with a MYSQL centralized database and an Apache server. Figure 8 depicts the global architecture of this prototype. In the following, we will discuss the main functionalities of Wiseshare collaborative system and their implementation in the prototype.

Accessing to the functionalities depends on the profile of the user. As we mention previously in the paper, we have three profiles of users: novice, expert and vandal. For the sake of implementation, a public user profile is needed; it represents new users exploring the system before registration or users that are not interested by contributing. The functionalities of each user profile are the followings:

- Public users can access the published contributions by searching them by specific criteria (by topic, con-

```
Create(u: user,c:contribution,e: environment,
        name: String, topic: String, h: URL)
{
  c.vis=created;
  c.idc=name;
  c.orig=u;
  c.chf=u;
  c.th=topic;
  c.d_cr=e.date;
  if u.rep= "novice"
    then{
        c.vis="restricted";
        }
    else{
        // c is created by an expert
        c.vis="published";
        }
  // pointer to an html page;
  c.ctt=h;
  Notify all the experts of the list E.Exp
  related to topic to check c;
}
```

Fig. 3. Effect of creating a contribution

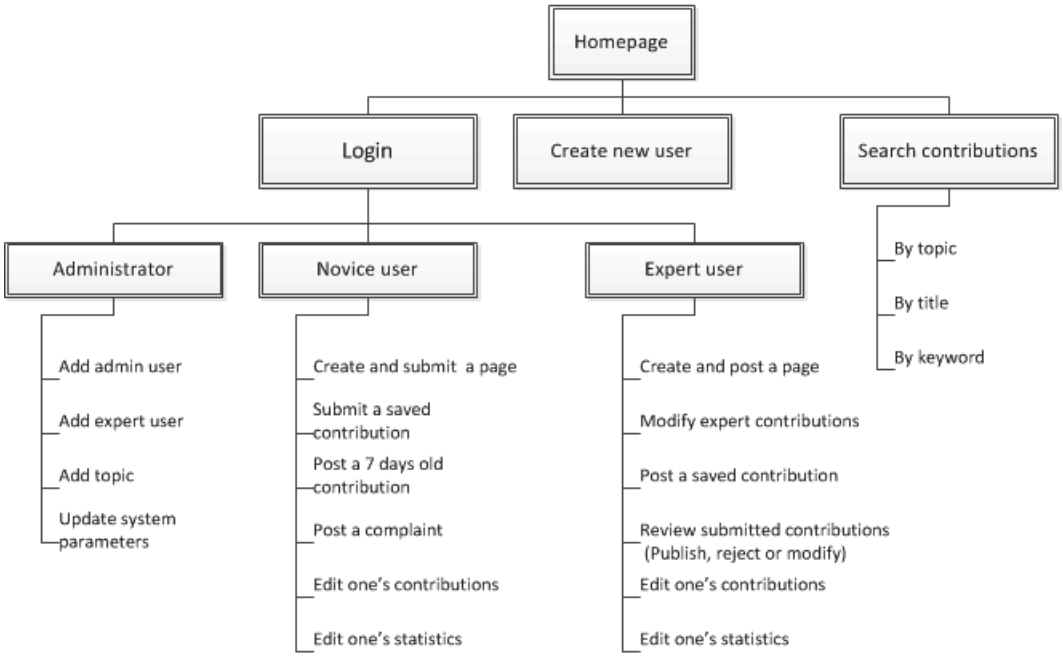

Fig. 8. The global architecture of WiseShare's prototype

```
Post(u: user,c: contribution)
{
 if (c.vis="restricted" AND c.chf== u.idu)
        OR (u.rep= "expert")
    then {
    c.vis="published";
    if  u.cb.(c.th) < 500
        then
            u.cb.(c.th) ++;
    if  u.cb.(c.th) == 500 AND u.rep= "novice"
        then
            u.Skl=u.Skl+c.th;
    if u.rep="novice" AND u.Skl is not empty
        then
            u.rep="expert";
   }
}
```

Fig. 4. Effect of posting a contribution

```
Edit(u: user,c: contribution,h: URL)
{
 if major(c.ctt,h)
    then   //c is drastically modified
        c.chf=u;
 c.ctt=h;
 Post(c.chf,c);
}
```

Fig. 5. Effect of editing a contribution

```
Suppress(c: contribution)
{

 if c.vis= "published" then
   c.chf.cb.(c.th)--;
 c.vis="suppressed";
 if c.chf.rep="expert" AND c.chf.cb.(c.th)==450
    then
     {
      retrieve (c.th) from the list of
      expertise of c.chf.tag;
      If c.chf.Skl is empty
        then
            c.chf.rep="novice";
     }
}
```

Fig. 6. Effect of suppressing a contribution

tribution title or keyword) and, in case they want to contribute, they should first register by creating a user account.

- Each new contributor is tagged as 'novice', novice users have access to the following functionalities:
  - Create a contribution: He can create a contribution in a specific topic (see figure 9), save it for further modifications and then submit it for the appreciation of 'expert' users.
  - Publish a contribution: one week after a contribution being submitted, the user can decide

```
Report(u_1: user,u_2: user, e: environment)
{
 u_1.Vdl=u_1.Vdl+u_2.idu;

 u_2.cpl++; // u_1 is a trustable complainer

 if (u_2.rep="novice" AND u_2.cpl=20) OR
    (u_2.rep="expert" AND u_2.cpl=100)
    then
        e.BL=e.BL+u_2.idu;
}
```

Fig. 7. Effect of tagging a user as vandal

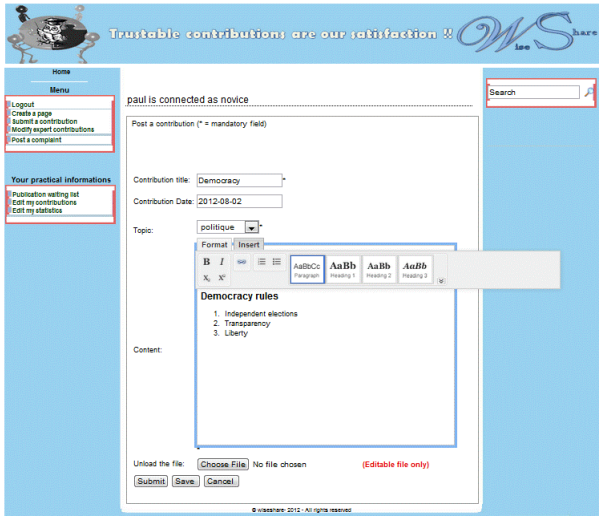

Fig. 9. Screenshot of contribution creation window.

to publish this contribution or wait for expert appreciation.
- Post a complaint against any user having a suspicious behavior.
- Edit their contribution statistics (see figure 10).

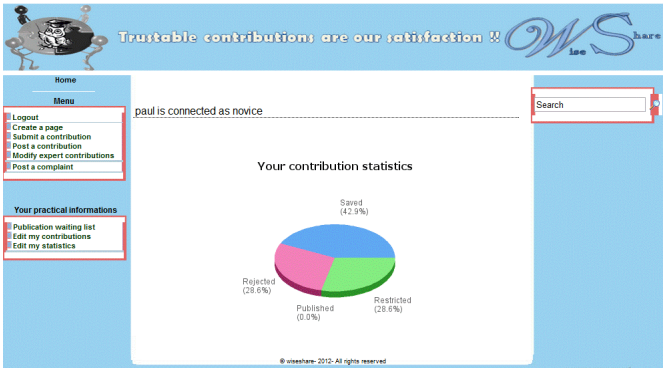

Fig. 10. Screenshot of contributions statistics window.

- Expert users have access to the following functionalities:
  - Create and post a contribution. They can also save their contributions for further modifications before publishing them.
  - Review novice users' contributions that are related to their area of expertise. They can publish them with or without modifications and can also reject them.
  - Post a complaint against any user who considered one of his actions as suspicious.
  - Review other experts' contributions that are related to their area of expertise.
  - Edit their contribution statistics.
- Vandal users cannot have access to the system through authentication. Thus, they become public users. However, they can no longer contribute to the system because they belong to the black list.

The implemented prototype of our system has the following advantages:

- Flexible and easy to use: It allows anyone to contribute to the topic he likes provided that he creates an account. Users are endowed with a simple environment for their contribution creation. They can use formatting tools as they wish. The statistics of user's contributions are edited using pie charts.
- Reliable: Unlike other systems such as Wikipedia, user's contributions are verified by experts before being published so that the contents are more reliable. This evaluation could bring contributors to be more serious in the contents they are producing in the system.
- Dynamic: In our system, the more a user is active by creating reliable contributions, the faster he becomes an expert user. Inversely, an expert user can also become a novice if he falls on creating unreliable contributions.
- Interactive: When a novice user submits his contribution, an email is automatically sent to all the experts in the concerned topic. Access to unauthorized functionalities is denied and error messages are displayed to inform the user. That allows those experts to be aware of waiting contributions so that they can evaluate them for publication. Our system also displays a warning when the number of a user's complaints is close to a predefined threshold.
- Automatic: An action taken in our system may automatically change the user's profiles or the object attributes.

VIII. Conclusion and Future Work

In this paper, we have proposed an attribute-based access based approach for creating and sharing knowledge in collaborative environment. In our system WiseShare, the

security model ensuring the reliability of the broadcasted information, is tightly woven. Indeed, each contribution has to obey to a strict access control policy. For this purpose, a contributory profile is built for each user during his activities. This profile reflects as faithfully as possible the expertise and the behavioral history of each contributor. Based on this profile, only enforceable requests are granted. Hence, a user may only perform contributions for which he is skilled. Our ABAC rules guarantee a responsible collaboration and minimizes clumsy contributions and vandalism actions. To achieve this goal, we have defined a formal framework to capture the features of WiseShare components. This formalism permitted us to specify our collaborative requirements. In addition, we have expressed the impact of each contribution on updating WiseShare profiles. In the future, we plan to integrate time constraints in experts profiles. This will encourage them to propose novel contributions in order to maintain their reputation. We intend also to investigate people-tagging mechanism to take advantage of the opinion of those experts who have effectively demonstrated their integrity throughout their contributions. WiseShare may probably lose in performance due to the centralized characteristic of the database. In terms of perspectives, we plan to integrate a decentralized management in our collaborative system. This feature will allow many experts to simultaneously work on the evaluation of some contribution. We will also measure the incidence of focusing on security concerns with respect to the flexibility of contributions. Finally, we intend to implement the mobile version of our system.

References

[1] T. Jaeger and A. Prakash. Requirements of role-based access control for collaborative systems. In *Proceedings of the first ACM Workshop on Role-based access control*, page 16. ACM, 1996.

[2] R.S. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman. Role-based access control models. *Computer*, 29(2):38–47, 1996.

[3] R.K. Thomas and R.S. Sandhu. Task-based authorization controls (tbac): A family of models for active and enterprise-oriented authorization management. *Database Security*, 11:166–181, 1998.

[4] R.K. Thomas. Team-based access control (TMAC): a primitive for applying role-based access controls in collaborative environments. In *Proceedings of the second ACM workshop on Role-based access control*, pages 13–19. ACM, 1997.

[5] C.K. Georgiadis, I. Mavridis, G. Pangalos, and R.K. Thomas. Flexible team-based access control using contexts. In *Proceedings of the sixth ACM symposium on Access control models and technologies*, pages 21–27. ACM, 2001.

[6] W. Zhou and C. Meinel. Team and task based rbac access control model. In *Network Operations and Management Symposium, 2007. LANOMS 2007. Latin American*, pages 84–94. IEEE, 2007.

[7] Covington, M.J., Long, W., Srinivasan, S. ,Dev, A.K. , Ahamad, M. and Abowd, G.D. Securing context-aware applications using environment roles. In *Proceedings of the sixth ACM symposium on Access control models and technologies*, pages 10–21. ACM, 2001.

[8] J. Kolter, R. Schillinger, and G. Pernul. A privacy-enhanced attribute-based access control system. *Data and Applications Security XXI*, pages 129–143, 2007.

[9] Waleed W. Smari, Jian Zhu, and Patrice Clemente. Trust and privacy in attribute based access control for collaboration environments. In *Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services*, iiWAS '09, pages 49–55, New York, NY, USA, 2009. ACM.

[10] Q. Wang, H. Jin, and N. Li. Usable access control in collaborative environments: Authorization based on people-tagging. *Computer Security–ESORICS 2009*, pages 268–284, 2009.

[11] P. Nasirifard, V. Peristeras, and S. Decker. An annotation-based access control model and tools for collaborative information spaces. *Emerging Technologies and Information Systems for the Knowledge Society*, pages 51–60, 2008.

[12] Y. Demchenko, L. Gommans, A. Tokmakoff, and R. van Buuren. Policy based access control in dynamic grid-based collaborative environment. In *Collaborative Technologies and Systems, 2006. CTS 2006. International Symposium on*, pages 64–73. IEEE, 2006.