User Centric Trust-based Access Control Management for Ubiquitous Computing Environments

Bo Fu, Declan O'Sullivan
School of Computer Science & Statistics, Trinity College Dublin, Ireland
{bofu, declan.osullivan}@cs.tcd.ie

Abstract — **Given the diversity of information and services that an individual will want to offer and/or share in a ubiquitous computing environment, it is critical that the individual is provided with mechanisms to manage access and usage to their resources. Mirroring real life, a natural concept for people to grasp when assigning ability to access or use resources is one of "trustworthiness". In previous work, it has been shown that trust is a multi-faceted and personal concept, and so this needs to be catered for in any trust management interactions an individual may have with the ubiquitous environment. This paper describes our recent research in examining such user interactions through the design, implementation and evaluation of an online social network application, as this application raises the kind of issues that will be faced by most individuals in ubiquitous computing environments.**

Keywords - trust; online social networks; multi-faceted model of trust; personalisation; ratings.

I. INTRODUCTION

Trust with broad definitions and concepts, somehow, works mysteriously. For many years, in various disciplines such as psychology, philosophy and sociology, researchers have tried to understand what trust means and how it works. As "ancient social traditions were designed to elicit trust during uncertain encounters, handshaking demonstrated the absence of weapons; clinking of glasses evolved from pouring wine back and forth to prove it was not poisoned" [37], the question becomes how can ubiquitous computing systems be designed to support users' management of online demonstrations of trust? Just as in real life, the representation and management of trust needs to be simple and personalised.

In our initial exploration of this question, we examined how access could be controlled by an individual to his or her location information in a decentralised instant messaging application [34]. The location information in question for this application was fairly simple and the trust decision uncomplicated but it did afford us the opportunity to test an underlying multi-faceted trust model that had been developed. However, it only provided limited opportunity to explore the individual's interaction with the trust system in personalising trust and in assigning degrees of trustworthy attributes across a range of types of information, ranging from information that should be public to sensitive information.

This paper reports upon our recent work that investigates the user centric trust management interaction issue in more depth. The application chosen to underpin our investigations is an online social networking application. This application was chosen for a number of reasons:

- it provides us with a rich set of diverse information/resources that a user has a personal interest in protecting, as will be the case in ubiquitous computing environments;
- its simple method of management interaction for a user is typically what we would want to emulate in a ubiquitous computing environment;
- it provides us with an opportunity to explore trust based access to more than a user's information resources but also potentially ubiquitous computing services that the user may offer in an open, structured or ad hoc way;
- and finally, trust based management of information by users in online social networks is currently very limited and this offers an opportunity to explore whether our approach might be useful in this domain.

This paper first provides some background and related work in section 2. In particular it provides an overview of the multi-faceted model of trust used in our work. Section 3 discusses the design of the user centric trust management interaction that we have incorporated into our online social network prototype named *miniOSN*. The implementation of *miniOSN* is outlined in section 4. Section 5 describes our evaluation, the scenarios used are first introduced followed by a comparison of *miniOSN* trust management features with a popular online social network - *Bebo*, results of evaluation interviews are also presented and analysed. Finally, section 6 provides some conclusions and future work.

II. BACKGROUND AND RELATED WORK

A. Trust Definitions and Characteristics

Trust has many definitions with subtle differences being expressed, therefore it is very difficult to fix upon what the term "trust" means. Trust is an elusive notion that stands for a diversity of concepts depending on the person involved. To some, trust is about predictability, where evidence of one's reputation suggests a most-likely outcome; to others, trust is about dependability, where one truly believes in and depends upon another; yet, to many, trust is simply letting others make decisions for you and knowing that they would act in your best interest.

The Cambridge Dictionary [6] presents a collection of definitions of trust, stating that to trust is to believe, trust is "the belief that you can trust someone or something"; trust can also be an arrangement, "in which a person or organisation controls property and/or money for the benefit of another person or organisation"; and at other times, to trust, means "to hope and expect that something is true". The Oxford Dictionary [32] states that trust is the "firm belief in the reliability, truth, ability, or strength of someone or something"; it is also the "acceptance of the truth of a statement without evidence or investigation".

In the computer science domain, Grandison and Sloman [21] defined trust as "the firm belief in the competence of an entity to act dependably, securely, and reliably within a specified context." Mui et al. [29] defined trust as "a subjective expectation an agent has about another's future behaviour based on the history of their encounters." Olmedilla et al. [31] stated that "Trust of a party A to a party B for a service X is the measurable belief of A in that B behaves dependably for a specified period within a specified context (in relation to service X)."

A quick glance of the definitions above present various terms and concepts such as belief, confidence and reliability, which once again indicate the difficulty in stating a definitive description for trust. In our research, instead of concentrating on a single definition, we focus on the characteristics of trust that hold true regardless of how trust is defined.

Studies [18] [8] of trust properties show that *trust is Asymmetric.* Levels of trust are not identical between two parties, in other words, A and B may not have the same amount of trust for each other. *Trust may be transitive*. Think of a situation where A and B know each other very well and are best friends, B has a friend named C whom A has not met. However, since A knows B well and trusts B's choices in making friends, A may trust C to a certain extent even though they have never met. Let us say that C has a friend named D whom neither A nor B knows well, A could find it hard to trust D. Hence, it is reasonable to state that as the link between nodes grow longer, trust level decreases. However, [20] [1] disagree and argue that trust is not transferable. This argument is exposed by [41], if "I have a friend who I trust not to lie. He is a gullible person who trusts the President not to lie. That does not mean I trust the President not to lie. This is just common sense."

Trust is personalised. Trust is subjective, two parties can have very different opinions about the trustworthiness of the same person. For example, a nation may be divided into groups who strongly support the political party in charge and groups who would strongly disagree.

Finally, *trust is context-dependent.* Trust is closely associated with overall contexts, in other words, trust is context-specific [23]. One may trust another enough to lend that person a pencil, but may find the person hard to trust with a laptop for instance.

B. Online Social Networks (OSNs)

The concept of assisted social networking dates back to the 1930s, when Vannevar Bush first introduced his idea about "memex" [38], a "device in which an individual stores all his books, records, and communications, and which is mechanised so that it may be consulted with exceeding speed and flexibility", and predicted that "wholly new forms of encyclopedias will appear, ready made with a mesh of associative trails running through them, ready to be dropped into the memex and there amplified."

Nowadays, Online Social Networks (OSNs) serve distinct purposes and focus upon various themes, that are mostly social, business, education, and entertainment oriented [14]. To date, there are hundreds of online social networking sites in Europe alone. Studies [24] have shown that in July, 2007, ranked by market share of visits across all industries, the most popular websites based on U.S. Internet usage was *MySpace* [30], ahead of *Google* [16], *Yahoo* [40], *Hotmail* [25] and *eBay* [10]. The popularity of OSNs is undeniable, and like most new technologies, such a young and exciting online social networking phenomena with rapidly growing communities welcomes innovation.

C. Computing Trust in Online Social Networks

Trust management is one of the most pressing issues in computer science, leading to various algorithms, systems and models being produced, such as PGP [42], REFEREE [7], SULTAN [22], FOAF [9], TRELLIS [15], Jøsang's trust model [27], Marsh's trust model [28], and so on. [19] provides a review of several notable trust projects with respect to their limitations for use in the OSN scenario. In addition, the authors proposed and evaluated algorithms that produce accurate trust values. However, most of these studies take a narrow and simplified view of trust that ignore the subjective nature of trust and assume a fixed definition in their models. For example, Golbeck [18] believes that "for trust to be used as a rating between people in social networks, the definition must be focused and simplified".

The drawback of such approaches is that they limit subjective expressions of trust and restrict the freedom for personalised views on trust within OSNs. Rather we believe a multi-faceted model of trust is required.

D. Quinn's Multi-faceted Model of Trust

Quinn [33] stated in his literature review that current trust management methods "tend to use a single synonym, or definition in the use of trust… such approaches can only provide a generic, non-personalised trust management solution". To address this problem of the lack of potential for personalising trust management, a multi-faceted model of trust that is personalisable and specialisable was proposed, implemented and evaluated. In his model, trust is divided into concrete concepts and abstract concepts with attributes of their own, where the former includes credibility, honesty, reliability, reputation and competency attributes, and the latter with belief, faith and confidence attributes. Ratings are then given to each of the eight attributes, and trust is calculated as the weighed average of these ratings.

The multi-faceted model of trust is outlined as the *Model of Trust* and separated across four models [33, p.50], as Figure 1 [33, p.48] shows. The *"myTrust Management Service* utilises a personalised model of trust, a domain specific model of trust, associated trust data, and trust policy to provide trust based recommendations to applications that operate in Internet environments." [33, p.49]

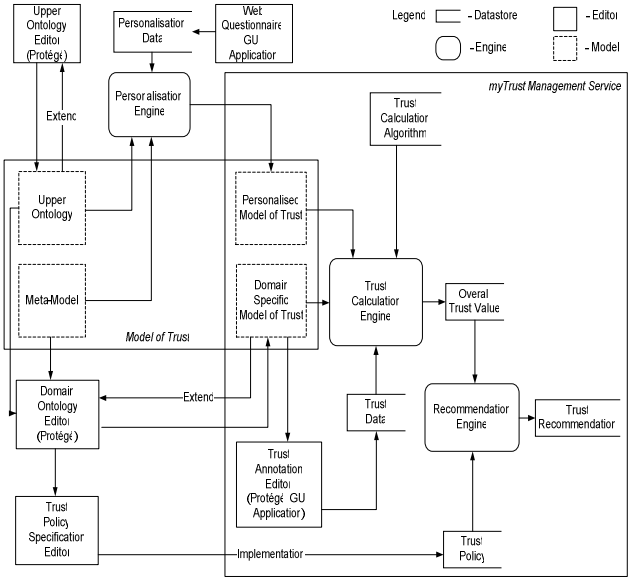

Figure 1. Trust Calculation Overall Framework

The upper ontology provides a set of trust concepts that are used in the generation of personalised models of trust and are also used to engineer specialised models of trust. The relationships that can exist between the extensible set of trust concepts is governed by the trust meta-model. A domain specific model is the instantiation of the upper-model and meta-model towards a given application domain. In domain specialisation the trust concepts in the upper ontology are sub-classed and domain specific properties are added. Domain models are kept separate to allow developers to capture and scope a range of domains, which can be used independently in applications. Personalised models of trust are generated from the upper ontology and meta-model on a per user basis. A personalised model contains the set of relationships that may exist between trust concepts as provided by an individual [33, p.50].

The claim for this model is that it has "the ability to capture an individual's subjective views of trust, also, capture the variety of subjective views of trust that are exhibited by individuals over a large and broad population", which in turn, provides "a tailored and bespoke model of trust". In addition to demonstrating its personalisation capabilities, Quinn demonstrated how the model could be specialised to any application domain.

The two applications that were used to trial the model and approach were web services composition and access control in a ubiquitous computing environment. However, Quinn did speculate in his conclusions that the model would be suitable for use in the OSN domain.

III. APPLYING QUINN'S MULTI-FACETED MODEL OF TRUST TO AN ONLINE SOCIAL NETWORK

In order to investigate whether Quinn's personalised approach to trust would be welcomed by users if incorporated into an OSN, *A Survey of Online Social Networks* was conducted [14]. Feedback on the multi-faceted model of trust was generally positive and we proceeded to design a prototype online social network, called *miniOSN*, to incorporate the multi-faceted model of trust and provide the users with a means to customise their subjective views of trust. In *miniOSN*, trust is defined with eight trust attributes, credibility, honesty, reliability, reputation, competency, belief, faith and confidence. And ratings can be given to all eight trust attributes depending on how users view trust. However, weighted average ratings of the eight trust attributes is removed in our design for the following reasons.

Predefined equality. If a weighted average is to be calculated of the eight trust attributes, it is presumed that these attributes are equally as important as each other, which conflicts with our goal of allowing personalised model of trust.

Suggested comparison. If each individual is associated with a score, comparisons of these scores may become unavoidable. However, in our model, scores are simply representations of subjective views on the trustworthiness of an individual. Comparing two people's average ratings could be misleading. For example, two people may have the same weighted average rating, however, one has high reliability rating and low honesty rating, while the other has high honesty rating and low reliability rating, how

could we conclude that one can be trusted more than the other? It is likely for such a rating feature to work well in an e-market application such as *Amazon* [2] and *eBay* [10] where users have no previous connections with one another, and are building relationships from scratch. However, in the OSN environment, this is certainly not the case. Most users of OSNs are friends with one another, before continuing to build their friendships online, where rating a person they already know well may become difficult. Findings from the initial survey also support our suspicion [14], hence, in our design, we have decided to make the ratings given to connected friends only viewable to the person who have rated them.

miniOSN has the basic functionalities that are commonly found among OSNs, it allows users to create accounts for themselves with a username and password along with a valid email address. Users of *miniOSN* can then set up representations of themselves, upload photos, post blog entries, as well as leaving comments in connected friends' profiles.

Fictional characters of the situational comedy *Friends* [11] were added as users of *miniOSN* to be able to study trust features and functionalities. The trust management mechanism designed in *miniOSN* aims to capture characteristics of trust as discussed in section 2.1.

miniOSN allows users to set trust rating requirements from a scale from one to ten, if desired, for each uploaded photo, blog and comment, as Figure 2 shows an example of the trust attribute matrix for a blog entry. Before posting a blog, users can specify required trust rating values for this blog. Similarly, by adjusting required values of the eight trust attributes before uploading a photo or leaving a comment, users can decide which friend(s) can view them.

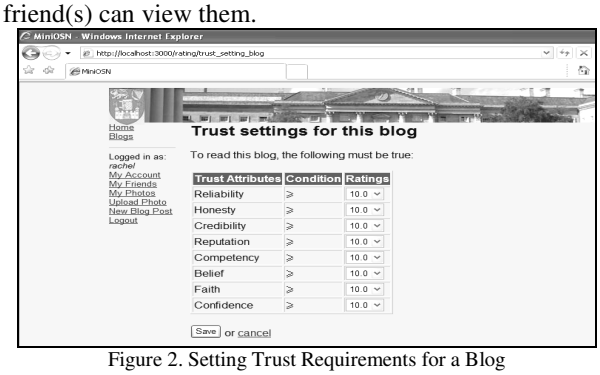

Figure 2. Setting Trust Requirements for a Blog

The trust features in *miniOSN* are presented in the following discussions.

Each user holds ratings of his/her connected friends in the database. Trust is personalised, two people can have different opinions of the trustworthiness of the same person, as we have found in section 2.1. To be able to capture this property of trust, in *miniOSN*, each user holds ratings of each one of their connected friends in the database, identified by *user_id*. For instance, in Figure 3, user Rachel has five connected friends in her profile, each "*Current Ratings*" link will then bring her to the page where that friend's current trust ratings are shown. As Figure 4 shows the list of Ross's current trust ratings according to Rachel, which are a different set of values than the ones Chandler has given Ross as Figure 5 shows.

By letting each user hold records of connected friends also enables users to express trust asymmetrically. For example, as Figure 4 shows, Rachel has a rating of nine for Ross's honesty attribute, Ross, on the other hand, rates Rachel's honesty as ten, as Figure 6 shows.

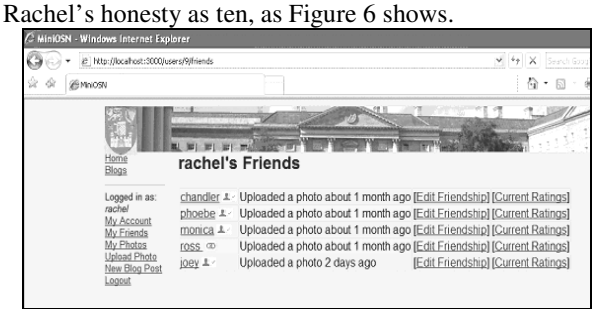

Figure 3. Rachel's Connected Friends

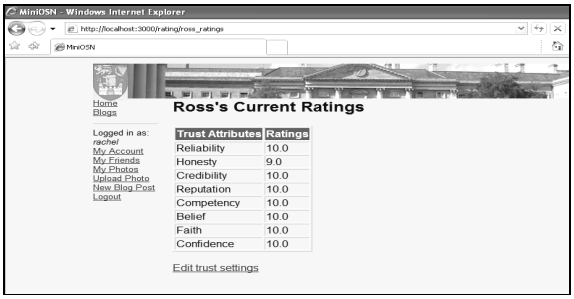

Figure 4. Rachel's Current Trust Ratings for Ross

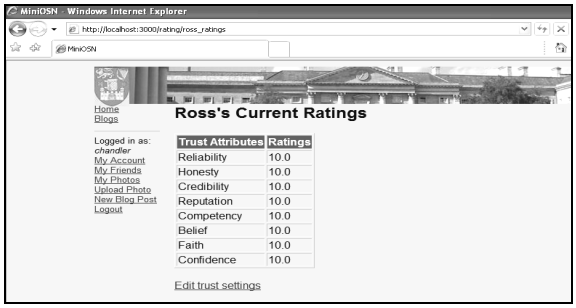

Figure 5. Chandler's Current Trust Ratings for Ross

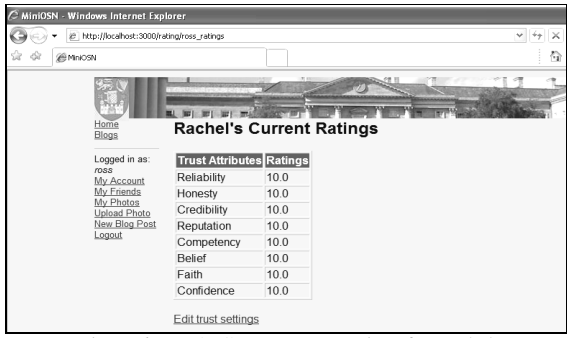

Figure 6. Ross's Current Trust Ratings for Rachel

The owner of a resource is able to set the trust requirements before distributing that resource. In *miniOSN*, whether it is uploading a picture, posting a blog entry or leaving a comment in someone else's profile, as long as you own this resource, you can set the required trust rating values for this resource. For example, in Figure 7, user Rachel can click on the "*Set Trust Rating Requirements"* link before uploading a picture in her profile, which will then direct her to the page where the trust rating matrix is, as Figure 8 shows, and she can then change the values of them against the ratings she has given her connected friends in order to grant different levels of access control to the picture she is about to upload.

Figure 7. Photo Upload Page

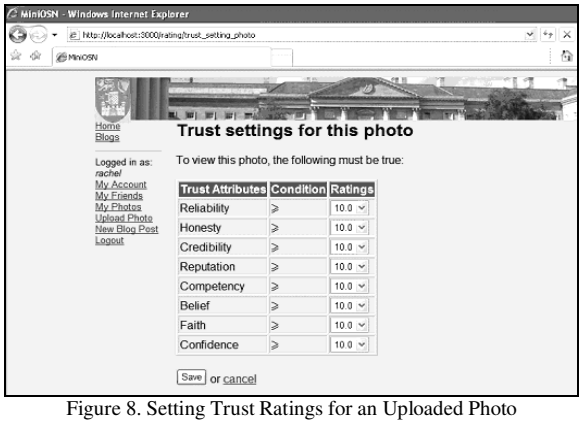

Figure 8. Setting Trust Ratings for an Uploaded Photo

All users and resources have default ratings of 10 out of 10. Some users may find rating a friend difficult, therefore, they can choose not to use the implemented trust rating feature, by simply ignoring the trust rating values since all are set to default values of ten out of ten, which means that all connected friends can access all resources in a profile, until a user makes changes to trust ratings of connected friends and/or trust rating requirements for certain resource(s) in the profile.

Users decide whether to transfer trust values to other friends of a connected friend. In *miniOSN*, users decide whether they would like to express trust transitively. In a user profile, once a connected friend's trust ratings have been set, the owner of the profile can then decide whether the same set of ratings should be transferred to all other friends of this certain friend, whom the owner of the profile is not currently connected to. For example as Figure 9 shows, once Rachel chooses option *"Yes"* to the question *"Would you like to apply these values to Ross's other friends?"*, all other friends of Ross whom Rachel is not connected to would have the same trust level as Ross.

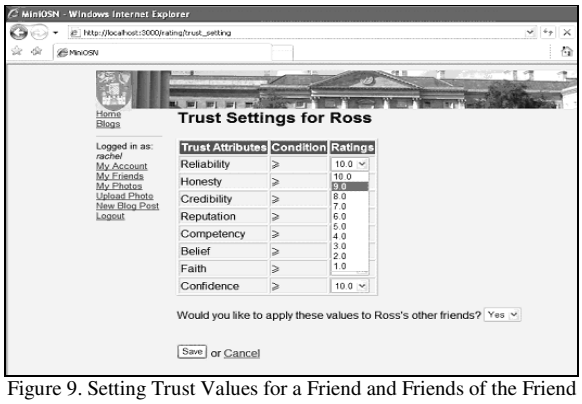

Figure 9. Setting Trust Values for a Friend and Friends of the Friend

Users decide the starting trust values of their connected friends. Although by default, all ratings are set to ten, however, the owner of a profile can adjust these settings and decide on whatever ratings they would like their friends to start with. Hence, the freedom of expressing various levels of trust among connected friends is provided in *miniOSN*.

Trust ratings can be reset whenever it is desired. In order to allow users to express trust context-specifically in *miniOSN*, the owner of a profile can change trust ratings for their connected friends whenever it is desired, depending on the situation. For example, if a certain blog should not be seen by a particular friend, the owner can adjust trust ratings of that friend so the blog in question is not accessible by that person.

IV. IMPLEMENTATION

Written in Ruby [36], *miniOSN* uses the Ruby on Rails [35] framework, running on Instant Rails 1.7 [26] for Windows. Influenced by [5], it uses plugins and techniques such as the Attachment_fu plugin [3] and XHTML Friends Network Microformat [39]. *miniOSN* follows a model-view-controller (MVC) architecture, separating data models, user interface and control logic of the application. In a Rails application, the model, holding all business logic, is a smart domain object that knows how to persist itself to a database. The View is a template that is responsible for generating the user interface, it simply inserts pre-built data between HTML tags. The Controller handles incoming requests by manipulating the Model and directing data to the View. As Figure 10 shows, *miniOSN* uses the MVC architecture powered by Rails, handles incoming requests from the client by sending HTTP requests to the Mongrel server, which then

forwards the request to the router, the router then finds the appropriate controller that will interacts with the model, the model then sends queries to the MySQL database and receives data/error from the database before responding back to the controller, which then invokes the view, telling the view to prepare XML, XHTML and CSS files for the data, and finally, the view sends back the representation of the data to the browser.

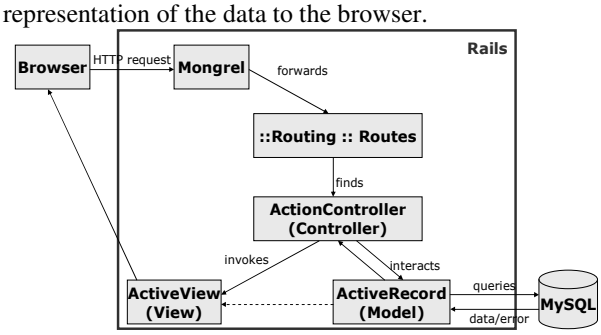

Figure 10. *miniOSN* Architecture

Incoming requests are first sent to a router, which then works out where in the application the request should be sent, and how the request itself should be parsed. Ultimately, this phase identifies a particular method, i.e., "action", somewhere in the controller code. The action might look at data in the request itself, might interact with the model, or it might cause other actions to be invoked. Eventually, the action prepares information for the view, which renders something back to the user.

Rails applications use an Object/Relationship Mapping (ORM) library, namely *ActiveRecord* to map the data stored in a database to a class in an application. *ActionPack* provides the view and the controller of the MVC stack: the View part of *ActionPack* creates the web pages while the Controller part holds the application together. For example, to manage user accounts [5], we first create a database migration file through *ActionRecord* as Figure 11 shows. So that the *Users* database holds the values of usernames, email addresses passwords, user profiles and maintenance information.

```
class CreateUsers < ActiveRecord::Migration
    def self.up
        create_table :users do |t|
            t.column :username, :string, :limit => 64, :null => false
            t.column :email, :string, :limit => 128, :null => false
            t.column :hashed_password, :string, :limit => 64
            t.column :profile, :text
            t.column :created_at, :datetime
            t.column :updated_at, :datetime
        t.column :last_login_at, :datetime
        end
        add_index :users, :username
    end
    def self.down
        drop_table :users
    end
end
```

Figure 11. *User Database* Migration Code Snippet

We then build an application controller, *UsersController* to display user profiles referenced by usernames, as the code snippets in Figure 12 shows. Finally, the views that correspond to the actions in *UsersControllers* are created, as Figure 13shows the code snippets of the *Sign Up* page and figure 14 shows the code snippets of the *Login* page.

```
class UsersController < ApplicationController
    def index
        @users = User.find(:all)
    end
    def show
        @user = User.find(params[:id])
    end
    def show_by_username
        @user = User.find_by_username(params[:username])
        render :action => 'show'
    end
    def new
        @user = User.new
    end
    def create
        @user = User.new(params[:user])
        if @user.save
            self.logged_in_user = @user
            flash[:notice] = "Your account has been created."
            redirect_to index_url
        else
            render :action => 'new'
        end
    end
    def edit
        @user = logged_in_user
    end
    def update
        @user = User.find(logged_in_user)
        if @user.update_attributes(params[:user])
            flash[:notice] = "User updated"
            redirect_to :action => 'show', :id => logged_in_user
        else
            render :action => 'edit'
        end
    end
end
```

Figure 12. *UsersController* Code Snippet

```
<h2>Signup</h2>
    <%= error_messages_for :user %>
    <% form_for :user, :url => users_path do |f| -%>
        <p>Username:<br /><%= f.text_field :username, :size =>
    40 %></p>
        <p>Email:<br /><%= f.text_field :email, :size => 60
    %></p>
        <p>Password:<br /><%= f.password_field :password,
            :size => 60 %></p>
        <p>Password Confirmation:<br />
        <%= f.password_field :password_confirmation, :size =>
    60 %></p>
    <p>Profile:<br /><%= f.text_area :profile, :rows => 6, :cols =>
            60 %></p>
    <%= submit_tag 'Sign Up' %>
<% end -%>
```

Figure 13. User Sign Up View Code Snippet

```
<h2>Login</h2>
    <% form_for :user, :url => {:action => 'authenticate'} do |f| -%>
        <p>Username:<br /><%= f.text_field :username, :size =>
    30 %></p>
        <p>Password:<br /><%= f.password_field :password,
            :size => 30 %></p>
        <%= submit_tag 'Login' %>
<% end %>
```

Figure 14. User Login View Code Snippet

miniOSN is constructed with fourteen action controllers such as *PhotosController*, *CommentsController*, *BlogsController* and *RatingsController*, where *RatingsController* invokes a user's trust rating database

which holds trust values of this user's connected friends and directs views of the rating matrixes.

V. EVALUATION

To evaluate the trust mechanism employed in *miniOSN*, two scenarios were created as discussed in section 5.1. We then compared the performances of *miniOSN* to that of the popular online social network *Bebo* [4] using the given scenarios, this comparison is presented in section 5.2. Finally, interviews were held to gather user opinions on the trust management approach developed, and these opinions are discussed in section 5.3.

A. Scenarios

The two scenarios were set with the following background [13]: Ross had three tickets to a New York Rangers game and wanted Joey and Chandler to go along, however Chandler noticed that if they did go to the game, they would not make it back in time for the Thanksgiving dinner which Monica hosted every year, therefore he tried to persuade the other two not to go themselves either. Although Ross and Joey agreed that it would be a bad idea to go, they secretly went to the game anyway.

In the first scenario, Joey took a picture at the game and decides to upload it to his online profile in *miniOSN*. With two connected friends Ross and Chandler to his profile, Joey knows that if he does not set trust rating requirements for the photo, Chandler would find out that they had gone to the game, however, the picture should not be a secret from Ross. Also, Joey can decide whether or not other friends of Chandler whom Joey is not connected to should see this photo. In this scenario, trust needs to be expressed asymmetrically as well as context-dependently, although Chandler has default trust ratings for Joey, Joey does not feel the same way about Chandler in this given situation. Trust is also personalised, since Ross and Joey would have different trust ratings for their mutual friend Chandler. Finally, trust can be expressed transitively since Joey decides whether Chandler's other friends should see the photo. Uploading a photo and posting a blog in *miniOSN* works similarly, but what happens when a comment is left in someone else's profile? In the second scenario, Ross posted a blog talking about the Rangers game days before the event, and Joey wants to leave a comment for Ross which concerns their meeting time, that should certainly be viewable by Ross since it is going to be left in Ross' profile. However, Joey may suspect that Ross trusted Chandler to see everything in Ross's profile. To prevent the comment from being viewable to Chandler, as the owner of that comment, Joey can then set trust rating requirements for this particular resource and stop Chandler from reading the comment.

B. Comparison of miniOSN and Bebo in the Scenarios

We used *Bebo* as the representation of notable OSNs in our study, due to its popularity with participants who took part in the initial survey [13, Figure 3-2], and registered these fictional characters in *Bebo*. We then compared performances of *miniOSN* to *Bebo* in the given scenarios. Since we are interested in modeling various degrees of trust subjectively among friends of a user, we say that Joey has set his profile "private" in *Bebo*, meaning that only people who are connected to him can see his profile.

Using *Bebo*, in scenario one, once Joey uploads the Rangers game photo, all of his connected friends would be able to see it. Joey is therefore, forced to grant Ross and Chandler with the same level of trust, even though in this situation, Joey does not trust Chandler as much as Ross. Currently, trust in *Bebo* can not be expressed asymmetrically, nor can it be tailored to a personalised view depending on the context. Moreover, since Joey has a private profile, Chandler's friend Monica therefore can not see Joey's uploaded picture, which means that trust is predefined as non-transitive. In scenario two, once a comment is left in Ross's profile, all of Ross's connected friends would be able to see it, meaning that both the owner of the comment: Joey, as well as Ross's connected friend Chandler can view the comment. In *Bebo*, there is no way for Joey to prevent that from happening.

The two scenarios showed that in *Bebo*, users can not express views on trust subjectively among their connected friends. Trust is assumed to be symmetric and non-transitive by the system, there is no notion of context-specific, nor is customisation provided.

With the first scenario in *miniOSN*, trust can be expressed asymmetrically depending on the context, where Chandler may have default trust ratings for Joey, Joey, on the other hand, does not trust his friend the same way in return. He was able to degrade Chandler's trust rating to restrict his access to the photo but not Ross. Not knowing the existence of such a photo, Chandler would not have felt being left out either.

In *miniOSN*, we could also portray a personalised view of the trustworthiness of an individual, as the second scenario shows. Ross trusts Chandler with all things in his profile, but Joey thinks otherwise and does not trust Chandler with a certain comment he had left in Ross's profile, by decreasing Chandler's trust ratings, he then prevents Chandler from reading the comment. In this situation, the system obeys Joey's trust requirements for the comment, not Ross's.

Finally, in *miniOSN*, users have the freedom of expressing their views on whether trust is transitive or not. Whether trust can be transitive or not is a personal preference, it is not fixed by the system. Joey can choose to let all other Chandler's friends have the same set of trust ratings, meaning that trust is transitive; or he can choose the option *"No"* when asked *"Would you like to apply these values to Chandler's other friends?"*, by which, he can then express that trust is non-transitive if he likes.

C. Interviews

The evaluation interviews were held on a one-to-one basis, with open-ended questions. There are three parts to the evaluation questionnaire, the first part aimed to find out whether there is the need to express various levels of trust among connected friends in an online social network. The second part concentrated on the created scenarios and gathered opinions on how well can users of *miniOSN* express their subjective views of trust asymmetrically, transitively, personally and context-dependently. Finally, the third part of the evaluation questionnaire focused on the trust management solution that is integrated in *miniOSN* and asked for suggestions of the refinement of its features. A total of nine volunteers including four Master students and three Ph.D. students in Computer Science, as well as two other non-technical candidates took part in our survey.

From the first part of the questionnaire, seven out of nine interviewees stated that they did not trust their connected friends equally, hence, they felt the need to express their various levels of trust among these friends in OSNs. However, two participants felt such a feature was not really necessary since they only used OSNs for irregular contact with people who are away. And most of all, they did not maintain resourceful profiles, as a result, they felt that there really was no need to distinguish one friend from another since nothing was meant to be a secret from anybody.

When asked whether they could relate to, or imagine situations where they wished they could have had a way to decide whomever friend to see whatever resources in their profiles whenever they want these friends to, except one person, eight candidates stated that such situations would be inevitable in OSNs they had experience with and given ways to have customisable control would be very helpful.

All of the interviewees felt that in the given scenarios, users could express trust asymmetrically and personally in *miniOSN*. However, candidates had contradictive views on whether trust can be expressed transitively depending on the context. Six candidates felt that this arguable characteristic of trust was not modeled well. Since users in *miniOSN* can only choose one of the two given options when asked whether they would like to apply a same set of trust ratings for all other friends of a connected friend, in other words, they can only state either they want to set the exact same trust level to every friend of the connected friend, or have no trust at all towards these people. This is an either-or approach, people who believe that trust level decreases as links between connected nodes grow longer can not express such a view in *miniOSN*.

Can trust be expressed context-dependently in *miniOSN*? Seven candidates felt that this is indeed the case. However, two participants disagreed, stating that the notion of trust is context-specific was not captured well. As once degrading a person's trust ratings would result in restricting this person's access to previous accessible resources in a profile.

Seven interviewees felt that such a trust rating feature in *miniOSN* would not hurt a friendship since the owner of the profile is the only person who could see the ratings, unlike findings in the initial survey where a large number of people [14] dislike the idea of rating each other's trustworthiness in OSNs.

We found that the eight trust attributes, reliability, honesty, credibility, reputation, competency, belief, faith and confidence were quite confusing to many. Several candidates mentioned that they found it hard to understand the concepts, and to distinguish them from each other was not easy since "not only do the attributes overlap with each other, but the differences among them are so subtle". Three people mentioned that they simply could not tell faith and belief apart, while two people felt that "competency" seemed out of place in an OSN environment, since it is very business like. Also, the number of trust attributes seemed to be overwhelming to many. Several candidates suggested that it may be possible to express trust with just three or four of the attributes instead of having all eight of them, which may help them see the differences in the concepts better and make full use of their understanding of them.

Finally, four interviewees felt that it was easy for them to associate numbers with the trust attributes, however, five other participants found it difficult to give numbers for them since understanding these concepts were so hard for them. Also, two people suggested using visual aid such as sliding bars for the rating system instead of using numbers.

Overall, seven out of nine volunteers felt that the trust mechanism implemented in *miniOSN* was helpful when expressing various degrees of trust, also it provide a better control over their resources in online profiles. However, it has been mentioned that the current rating system in *miniOSN* seems to be over-complicating the matter. Several candidates felt that it would be just as efficient and effective if users of a profile can simply specify which friend should see what resources without having to go through a rating matrix for every one.

D. Interview Conclusions

In summary, from the evaluation interviews, we found that users welcome trust management systems that allow expressions of subjective views on trust. However, refinement is necessary on the design and user interface in *miniOSN*.

Currently, in *miniOSN*, we are heavily relying on users to keep track of all trust ratings of their connected friends, which works well on a one-to-one basis, however, when one has to manage a large number of friends, it becomes difficult for the user to keep track of various sets of numbers.

Also, we could improve user interface by implementing functions where once a set of trust rating requirements has been set for a certain resource, users should then be notified with a list of connected friends who do have access to the resource, to avoid human error. It would also be convenient to let users see a list of all their connected friends and corresponding trust ratings on one page, for easy comparison and readjusting.

Our application of a multi-faceted model of trust addresses the problem of a lack of personalisation when modeling trust in OSNs, however, a common view of trust level being degraded as the link between nodes grow longer has not been captured well. Also, as mentioned earlier, an action such as degrading trust ratings of a person would result in restricted access to previous accessible resources, which clearly, is a major problem that needs to be solved.

Finally, it is necessary to take into account the limitations of our interviews, since only a small number of candidates were available for the evaluation experiment.

VI. CONCLUSIONS AND FUTURE WORK

The results from our initial experiments with users have been encouraging and useful in the identification of issues that need to be address:

Can we reduce the number of trust attributes displayed? As the findings in the evaluation interviews suggest, many candidates feel that the number of trust attributes are overwhelming. Instead of defining trust with eight attributes, could we reduce them to a total of four possibly? Alternatively, could we provide beginners with a simplified model of trust and an advanced version for advanced users who desire a level of sophistication? Answers to such questions require further surveys and experiments.

Can we choose collections of trust attributes and assign priorities to them? Some may argue that by reducing the number of trust attributes, we risk restricting ourselves to a limited design right from the start. Instead of cutting down on the number of trust attributes, another alternative is to let users specify their very own models of trust. For example, trust, in one person's opinion, can mean a combination of credibility, honesty, reputation and confidence, while for another, trust may stand for competency and reliability. As well as letting users to choose collections from the given eight trust attributes, we can also let users decide which attributes are more important than others, by assigning priorities to them. For instance, a friend's reliability is more important than his/her honesty in certain scenarios, while at other times, competency is valued more than reputation. By having such features, the application would have even more refined personalisation when modeling trust in the OSN environment, where users can tailor their needs by creating different models of trust in the same individual for different situations, and their various levels of trust as well as different kinds of trust among connected friends.

Can we add features that enable users to deal with groups as well as individuals? Such features should let users assign their connected friends into different groups with various trust values. Management of trust ratings would therefore become much easier. This would be advantageous in situations where a user is connected to a large number of people but not necessarily having to know all of them well, grouping these not-so-close "friends" and assigning the same trust values to the ones within a group may be highly beneficial.

Can we make trust ratings public? Our application has been hugely influenced by the findings from the initial survey [14], where a strong disliking of the rating feature was presented, mainly due to worries of hurting friendships and concerns of the possibility of encouraging online bullying behaviour. Therefore, in our design, we have decided to let ratings for others be viewable by the owner only. However, at the stage when we carried out our initial survey, participants had very little ideas of the functionalities of such a rating system in the context of online social networks and without any knowledge on the proposed solution, a sense of insecurity was developed. We foresee experiments and another application where ratings are visible to others. Comparison of the two approaches could bring us to interesting findings and conclusions.

Can we develop real world deployment and perform larger case studies? It would be interesting to gather a much larger audience for the evaluation of the proposed solution. There is certainly, a huge scope for the continuation of research in the area of trust management in OSNs, as a proxy for the kind of issues that need to be addressed for user centric ubiquitous computing applications. Case studies could be designed to tackle various issues discussed in this paper. Besides experimenting with online communities in the broad Internet context, it would also be very interesting to address issues with OSNs used in organisations and centre around particular user requirements in a work setting.

In summary, it is the opinion of the authors that trust-based management mechanisms that are tailorable to an individual's needs are necessary in many collaborative ubiquitous computing applications. In our research, we have shown that the interactions necessary with the individual in order to manage and personalise trust can be simplified but that more work is needed to fully explore the issues that arise from our initial evaluation.

REFERENCES

[1] Abdul-Rahman, A., "A Framework for Decentralised Trust Reasoning", Ph.D. Thesis, University of London, UK. 2004.

[2] Amazon homepage, http://www.amazon.com

[3] Attachement_fu plugin project website homepage, last retrieved from http://svn.techno-weenie.net/projects/plugins/attachment_fu

[4] Bebo homepage, http://www.bebo.com

[5] Bradburne, A., 2007, “Practical Rails Social Networking Sites”, Apress.
[6] Cambridge Learner’s Dictionary with CD-ROM, Cambridge University Press, 25 June 2007.
[7] Chu, Y., Feigenbaum, J., LaMacchia, B., Resnich, P. and Strauss, M., “REFEFEE: Trust Management for Web Applications”, the *World Wide Web Journal*, 1997, 2(3), pp. 127-139
[8] Dey, A., “Understanding and Using Context”, *Personal and Ubiquitous Computing*, 2001, 5(1): 4-7.
[9] Dumbill, E., “XML Watch: Finding friends with XML and RDF”, *IBM Developer Works’*, June 2002, last retrieved from http://www-106.ibm.com/developerworks/xml/library/sfoaf.html
[10] eBay homepage, http://www.ebay.com
[11] Friends Situational Comedy, 22 Sep., 1994 – 6 May, 2004, Bright, K. (producer), Crane, D. and Kauffman, M. (creators), United States: Warner Brothers.
[12] Friends Situational Comedy, Episode 226, Season 10, Gary Halvorson (director), 20 November, 2003.
[13] Fu, B., 2007, “Trust Management in Online Social Networks”, M.Sc. thesis, School of Computer Science and Statistics, Trinity College Dublin.
[14] Fu, B. and O’Sullivan, D., “Trust Management in Online Social Networks”, *7th IT&T Conference -Digital Convergence in a Knowledge Society*, October 25-26, 2007, ITB, pages 3-12.
[15] Gil, Y., and Ratnakar, V., ‘Trusting Information Sources One Citizen at a Time’, *Proceedings of the First International Semantic Web Conference (ISWC)*, Sardinia, Italy, June 2002.
[16] Google homepage, http://www.google.com
[17] Golbeck, J. A., 2005, “Computing and Applying Trust in Web-Based Social Networks”, Ph.D. thesis, Faculty of the Graduate School, University of Maryland.
[18] Golbeck, J., 2006, “Trust and Nuanced Profile Similarity in Online Social Networks”, *Journal of Artificial Intelligence Research*, MINDSWAP Technical Report TR-MS 1284.
[19] Golbeck, J. and Hendler J., 2006, “Inferring Binary Trust Relationships in Web-based Social Networks”, *ACM Transactions on Internet Technology*, Volume 6, issue 4, Nov. 2006, pages 497-529.
[20] Grandison, T., “Trust Management for Internet Applications”, Ph.D. thesis, University of London, UK. 2003.
[21] Grandison, T., and Sloman, M., 2000, “A survey of trust in internet applications”, *IEEE Communications Surveys and Tutorials*, 4(4):2–16.
[22] Grandison, T. and Sloman, M., ‘SULTAN - A Language for Trust Specification and Analysis’, *Proceedings of the 8th Annual Workshop HP Open View University Association (HP-OVUA)*, Berlin, Germany, June 24-27, 2001.
[23] Gray, E. L., 2006, “A Trust-Based Management System”, Ph.D. thesis, School of Computer Science and Statistics, Trinity College Dublin.
[24] Hitwise Data Centre, “Hitwise: One in 20 Web Visits Go to Social-Networking Sites”, 2006, last retrieved from http://www.marketingvox.com/archives/2006/11/09/hitwise_one_in_20_web_visits_go_to_socialnetworking_sites
[25] Hotmail Homepage, http://www.hotmail.com
[26] Instant Rails homepage, http://instanrails.rubyforge.org
[27] Jøsang A., 1996, “The right type of trust for distributed systems”, *Proceedings of the 1996 workshop on new security paradigms*. Lake Arrowhead, California, United States, ACM Press.
[28] Marsh S., 1994, “Formalising Trust as a Computational Concept”, Ph.D. thesis, Department of Mathematics and Computer Science, University of Stirling.
[29] Mui, L., Mohtashemi, M., and Halberstadt, A., 2002, “A computational model of trust and reputation”, In *Proceedings of the 35th International Conference on System Science*, pages 280–287.
[30] MySpace homepage, http://www.myspace.com
[31] Olmedilla, D., Rana, O., Matthews, B., and Nejdl, W., 2005, “Security and trust issues in semantic grids”, In *Proceedings of the Dagsthul Seminar, Semantic Grid: The Convergence of Technologies*, volume 05271.
[32] Oxford University Press, 23 Jun. 2005, Compact Oxford English Dictionary of Current English.
[33] Quinn K., 2006, “A Multi-faceted Model of Trust that is Personalisable and Specialisable”, Ph.D. thesis, School of Computer Science and Statistics, Trinity College Dublin.
[34] Quinn, K., Kenny, A., Feeney, K., Lewis, D., O’Sullivan, D. and Wade, V., “A Framework for the Decentralisation and Management of Collaborative Applications in Ubiquitous Computing Environments”, *10th IEEE/IFIP Network Operations and Management Symposium (NOMS 2006)*, Vancouver, Canada, 3-7 April 2006, Springer, 2006, pp 1-4.
[35] Ruby on Rails Plugins homepage, http://wiki.rubyonrails.org/rails/pages/Plugins
[36] Ruby Programming Language homepage, http://www.ruby-lang.org
[37] Shneiderman B., 2000, "Designing trust into online experiences". Commun. ACM 43(12): 57-59.
[38] Vannevar, B., 1996, "As we may think." interactions 3(2): 35-46.
[39] XHTML Friends Network microformat project homepage, http://www.gmpg.org/xfn
[40] Yahoo homepage, http://www.yahoo.com
[41] Zimmermann, P., “PGP(tm) User's Guide”, October 1994.
[42] Zimmerman, P.R., “The Official PGP Users Guide”, MIT Press, Cambridge, MA, USA, 1995.