POLICIES FOR DISTRIBUTED USER MODELING IN ONLINE COMMUNITIES

A Thesis Submitted to the College of

Graduate Studies and Research in Partial

Fulfillment of the Degree of Master of Science

in the Department of Computer Science

University of Saskatchewan

Saskatoon

By

Tariq Muhammad

Permission to Use

In presenting this thesis in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
University of Saskatchewan
Saskatoon, Saskatchewan, Canada
S7N 5A9

Acknowledgement

I am extremely thankful to my supervisor Dr. Julita Vassileva for her support in this thesis, for listening patiently my ideas and generously advising on research directions. I also want to express my gratuities to my committee members, Dr. Ralph Deters, Dr. Simone Ludwig, and Dr. Denard Lynch for their insightful comments and suggestions for my work and initial proposal. I am also grateful of other students at MADMUC lab for their help and support. I extend my special thanks to Ran Cheng for his efforts that he put in earlier version of Comtella Project and my orientation with application. I also want to thank my dear wife Kulsoom for all her support and unconditional love in all these years.

Contents

List of Figures

List of Tables

List of Acronyms

IRC	Internet Relay Chat
LDAP	Lightweight Directory Access Protocol
MADMUC	Multi-Agent Distributed Mobile and Ubiquitous Computing
OASIS	Organization for the Advancement of Structure Information Standards
OWL	Web Ontology Language
PEF	Policy Execution Framework
RBAC	Role Bases Access Control
RSS	Really Simple Syndication
SAML	Security Assertion Markup Language
UM	User Modeling
UMS	User Modeling Server
WWW	World Wide Web
XACML	eXtensible Access Control Mark-up Language
XML	eXtensible Mark-up Language
GUI	Graphic User interface
AJAX	Asynchrones Java and XML Technique

ABSTRACT

The thesis addresses three main problems in the area of user modeling and adaptation in the context of online communities:

1) Dealing with unique and changing user modeling needs of online communities.
2) Involving users in design of the user modeling process.
3) Interoperability of user models across different communities.

A new policy based-approach for user modeling is proposed, that allows explicit declarative representation of the user modeling and adaptation process in terms of policies, which can be viewed and edited by users. This policy-based user model framework is implemented in the MCComtella community framework, developed as part of this thesis work, which allows hosting multiple communities, creating new communities by users, and which supports users in setting explicit user modeling policies defining participation rewards, roles and movement of users across communities.

.

Chapter 1

Introduction

1.1 User Modeling in Online Communities

There are many definitions available in literature for online community. One of the most frequently cited definitions is by Jenny Preece, ‘a group of people, who come together for a purpose online, and who are governed by norms and policies’ (Preece, J., 2000). The basic purpose of online communities is to support social interactions and exchange of digital resources among people (Kimberly et al., 2003) (DeSouza and Preece, 2004). Internet based online communities enhance “the reach, depth and its potential impact” of groups that already exist in physical world” (Preece et. al. 2003). Online communities create a virtual space which enables users to store digital artefacts (web links, files and discussion threads), and maintain ownership of these artefacts. They allow the users to develop status and reputation and as a result, a sense of belonging to the online community. Online communities are virtual worlds but they have norms and polices which are derived in and out of the virtual world (Kimberly et. al. 2003). To resolve the possible conflicts with rights of belonging and ownership, online communities require user polices and user models.

User policies are a set of rules and regulations about the user behaviours in an online community. User policies can define user roles, set the access rights and visibility of digital artefacts, protect the privacy of users, and implement mechanisms for rewarding user participation.

Computer systems implement user models to manage information about the users in order to provide adaptations and personalized services. User models, as defined by Alfred Kobsa (Kobsa 1995, p.1) are "collections of information and assumptions about individual users (as well as user groups) which are needed in the adaptation process". Therefore a user model is more than a pair of user name and password - it is a detailed profile of user that can be used for adaptations in the user interface to achieve personalized interaction with the user. It can include user preferences, knowledge or experience, user roles, status, interaction history and others. Normally, user models are updated automatically by procedures that are deeply embedded in the application code. Once designed and implemented, these procedures cannot be changed, for example, to take account for new features of the user. Similarly, the adaptation mechanisms that are based on the user model are usually built in the system design and not changeable.

In online communities, however, the user modeling process and the corresponding adaptations may vary widely from one community to another, depending on the specifics of the community, the wishes of the owner / administrator, or the entire community. For example, how many and what kind of contributions does a user have to make to the community (say, in Wikipedia or Slashdot) to earn the rank "moderator" depends on the particular community. The existing statically built-in user models and adaptation techniques do not support well the variety and dynamics required; since they are deeply embedded in the code of the community infrastructure and require professional programmers familiar with the infrastructure to change them. The best solution would be to give control over these processes to the community owners/administrators and/or the entire community. But for this, it is necessary to open the user modelling and adaptation mechanisms – that is, make them understandable and editable for users. This approach would fit well with the Web 2.0 trend of involving end-users in running their communities.

One solution to achieve openness (transparency and editability) is to use policies to create and manage user models and the adaptation processes based on the user models. These policies are rules that govern the updating of the user model and the resulting adaptations. While policies may be applied to create and manage user models and adaptation in a single user application, they fit well intuitively in the context of a community, where they can express the norms and values of the community, just like laws express some of the norms in a country.   For example, it would be beneficial if the rules according to which the user can attain a certain status in the community based on her level of participation (user model) and the privileges associated with the status, are expressed in an open policy that is visible and can be discussed by the community. This is a novel idea that has not been explored yet in the area of user modeling or online communities and will be explored in this thesis.

Another problem of online communities that is addressed in this thesis is that current designs of online communities do support collaboration among online groups. Most existing communities are independent from each other; allow very little sharing and/or interaction across online community borders. Even though currently there are attempts to bridge across communities through mashups, the integration is very shallow – it allows users to switch between different communities within the same browser window typically without requiring them to log-in separately to every community (they have to do it only once in the beginning). However, there is no integration of the data from different communities; they are like “walled gardens”. Inter-community collaboration can help resolve this issue of participation and sustainability. One of the main design problems to ensure inter-community collaboration is the transfer of the user data, including the user identity and user model, across online communities. Even if designers of two web sites agree on exchange of contents, it is hard to transfer a user model across these sites, since the different online community applications typically use different database organizations or different ontologies and can therefore not transfer and understand the user model data received upon request from another application (community). The need arises for a mechanism to create a user model on request just in time according to the current context in new community. This thesis proposes that the policy-based user modeling and adaptation approach can be useful in addressing this problem.

In order to implement and evaluate this approach; a platform is needed. Current online communities allow users to share identity, but do not allow transferring other user attributes such as their roles in a community and status. There is need to develop a platform where users can create communities and participate in these communities while maintaining a separate user profile in each group. For this thesis we developed such a platform called “Multi Community Comtella” (MCC).

1.2 Main Problems and Proposed Approach

In summary, this thesis addresses several problems in the area of user modeling and adaptation in the context of online communities:

- Dealing with *unique and changing user modeling needs* of online communities
- *Interoperability* of deeper user models across different communities.
- *Involving users in the user modelling process*

To address these problems, this thesis presents an approach based on transparency for the user and editable policies for user modeling and adaptation. This user modeling approach is human centric in nature. It allows users in the community to view, discuss and ultimately, change the policies which define how the user model is computed, updated and how it is used for adaptation to individual differences. This approach was implemented and evaluated in specially designed community infrastructure called Multi-Community Comtella (MCComtella).

The thesis is organized as follows: Chapter 2 reviews the existing studies and solutions for user modeling systems. Chapter 3 gives an introduction of the history of the Comtella community framework and the design of the MCComtella, the platform used to implement and evaluate the proposed approach. Chapter 4 describes the proposed approach. It introduces the concept of policies within and across communities and provides examples of user policies in MCComtella. Chapter 5 describes the evaluation study of the approach in MCComtella and presents the results along with a discussion. Chapter 6 provides a summary of the thesis, conclusions and directions for future work.

Chapter 2

Background and Related Work

This chapter outlines the evolution of user modeling approaches leading to decentralized and open user models approaches applied in ubiquitous computing, social and learning environments.

2.1. Evolution of User Model and Application Adaptations

User models support adaptation and awareness in computer applications. A user model provides appropriate information for the application to adapt the functionality and interface according to the user's preferences, knowledge, or context. The need to adapt the functionality and data access rights to particular users emerged with the first multi-user systems back in 1960's. This lead to the creation to the first Role-Based Access Control (RBAC) systems, which have been used ever since. In role-based access control systems, users are associated with roles defined according to the operational needs of groups and organizations. Rights of access are defined at the role level. Users can work in one or more roles and can perform actions allowed to these roles (Mohammed and Dilts 1994, Sandhu and Park 1998, Park, Sandhu and Ahn 2001).

The RBAC approach has also been extended recently. Kagal et al. (2001) proposed an ontology-based RBAC approach for pervasive computing environments. This approach allows expressing not only user role hierarchies but also user properties in XML language. This ontology-based approach allows making changes in the rights of individual user without changing the role of user. Denaux et al. (2005) proposed an approach for interactive ontology-based user modeling to allow for interoperability, adaptability and overcome the “cold start” problem.

RBAC, however, can not implement the level of adaptation necessary in online communities. These online applications may require adaptations to a much larger set of possible group users, based on user goals, capabilities, user attitudes and knowledge, (Kass and Finn 1988). Classifying users according to all these dimensions create many possible combinations, impossible to cover with a fixed set of roles. User models, even simple stereotype models (Rich, 1979), allow handling the variety of users in more flexible way, to provide individual adaptation (Rich, 1983).

User models emerged in the late 1970ies in the area of natural language understanding / generation, as an effort to interpret utterances correctly. The first user modeling approach (Rich, 1979) was based on stereotypes of users (a stereotype is a labelled set of user parameters with particular value ranges). Based on certain ‘cues’ in the users’ interaction with an application, certain stereotypes is triggered, e.g. “educated-person” or “interested in sports”. Users would be classified under one or combinations of stereotypes, e.g. “intellectual, woman”, “interested in sports, businessman”, or “woman, feminist”. The classification can be used to adapt the system to the user, for example, to recommend books / movies, or to help the system better interpret ambiguous utterances of the user in the context of natural language dialogue. User models evolved further to incorporate complex representation of user characteristics and sophisticated updating mechanisms.

2.2. User Modeling Approaches

This section reviews various existing user modeling approaches that are relevant to online communities. We are focusing on approaches that deal with user models in decentralized environments to highlight the need of mechanisms discovery for user model fragments. We also point out the related works that focus on open, active and interoperable user models.

2.2.1. *Procedural and Declarative Approaches*

Two main approaches for user modeling evolved – a procedural (Brown & Burton 1978) and declarative (Schulungbaum & Ewert 1996).

In the procedural approach applications use a set of specialized procedures as user models; for example, a set of procedures that implements a correct and several different erroneous algorithms that children use to subtract 2-digit numbers. Based on the child's input (solutions to subtraction problems), the user modelling component classifies the child's method of solving problems into one of the procedural models programmed in the system. These models are hard to develop, since they require extensive and deep knowledge of the domain, but when accurate, they are very useful, since they allow predicting the user's behaviour. Unfortunately, they are too expensive to be practical. They are always application-specific (Anderson, 1988). Changes are not easy to implement, since the procedures are part of the application code.

The currently dominant user modeling approach is declarative approach, where facts can be added and removed or values are updated in a uniformly represented user model. The applications using this approach can use a standard reasoning mechanism to make conclusions based on the information in user models. The representation can vary depending on the reasoning mechanism used. There are declarative approaches based on vectors with values for certain variables (profiles), rules like those used by expert systems, probabilities used with Bayesian belief networks, or more recently, statements about the user, expressed according to agreed upon ontology and language, ensuring interoperability across applications. The declarative approach focuses on the expression of the user model instead of the processes (algorithms) for discovery, interpretation and integration of the user model data, since these processes are standardized in the chosen reasoning mechanism. The declarative models are modular in nature and can be easily executed (Niu 2004)

2.2.2. Centralized and Decentralized Approaches

With the emergence of the web, user model servers appeared allowing to offload the task of collecting and updating user data to centralized servers. UMS (Kobsa, A. and Fink, J. 2006) is a user modeling server based on Lightweight Directory Access Protocol (LDAP). Client applications can store and retrieve information about their users. UMS offers pre-defined and user-defined types, distribution across network, replication and synchronization and security by using known standards. However, these models are decontextualized – by storing the user data that comes from a particular application to a centralized server and user data representation scheme, context-specific data is lost.

Decentralized user modeling approaches (Vassileva et al, 2003)were proposed to deal with this problem, which gains importance in environments that are inherently decentralized , agent-based, mobile and ubiquitous and also in online communities. The active approach (McCalla et al. 2000), distributed (Vassileva, 2001), and purpose-based (Niu et al., 2004) are examples of the decentralized user modeling approach. This involves interpreting and integrating data obtained in different context and using possibly different schema. In the active user modeling approach, the model is computed just in time according to the current purpose and context user information from different sources (applications). This approach focuses on the computation of the model by discovery, retrieval, integration and interpretation of fragmented models rather than on mere knowledge representation. This model gets its name from the emphasis on the active computation of user models on demand (in contrast to utilizing information from the stored model). A more generalized version of the active approach emphasizes its decentralization (Vassileva et al., 2003). In this approach user information is scattered around in independent and autonomous agents as user model fragments. Each agent develops these fragments according to its context, adaptation purpose and uses its own representation schema. The agents can share user model information among each other, and they interpret it depending on their relationship with each other and the purpose at hand.

2.2.3. Open User Model Approaches

Most applications keep their user model hidden from the user. However, applications such as learning environments need an open learner (user) model, to allow reflection by the learner on what she knows and correction of errors in the model (Bull. and Pain 1995, Bull 1997). UM (Kay, J. 1995) is toolkit for user modeling, which helps the user to understand her own model. The system also stores the source and evidence of the user information. This toolkit provides a bridge to transfer the user’s and system’s beliefs in shared space and allows both user and system to view each other’s model.

Open user modeling approach typically uses a declarative representation and the emphasis of research is on finding appropriate way of presenting or visualizing this declarative representation to the user, so that she can understand it.

Open user / learner models often provide an interface allowing the user to correct the model if required. Of course, the user model should be protected from undesirable changes, for example, a learner who wants to correct the system’s representation of her knowledge to show full understanding, when she hasn’t demonstrated such understanding in her interaction with the system. In learning communities, an open and inspectable UM helps students and teacher in reflection and assessment. Both the learner and the teacher can collaborate in managing the UM (Zapata-Rivera & Greer 2001). The motivations behind making the UM visible and editable by the user also include providing user with awareness of why certain adaptations happened (Kay, J 2006). Bull and Kay (2005) summarize the reasons for the openness of the user model: insuring accuracy, enabling reflection, enabling planning/monitoring performance, enhancing awareness (about other users) in collaboration, providing user control and privacy - the user’s right to know the information that is stored in the system about her.

The active modeling approach can be combined with open user models to create small fragmented models just in time that can be inspected by the user (Hansen and McCalla, 2003). More generally, the decentralized user modelling approach supposes openness (Vassileva et al., 1999). Decentralized open models have been applied in the area of online learning environments, since a learner may use several online learning systems at a time to meet her learning needs. Such a learner can benefit from an open and interoperable user model, which offers methods of expressing the user profile and accessing it though a special interface (Dolog and Schäfer 2005). Dolog and Schäfer 2005 developed a framework for browsing, manipulating and maintaining interoperable learner profiles. This framework addresses the issues of fragmented user profiles in different learning communities. They suggest that the availability of multiple fragments of user profile across the different communities can benefit these learning applications in making decisions for adaptation. They provide a server implementation of such an exchange of interoperable leaner profile by using open standard and open API. It allows the access the Resource Description Framework (RDF) based user profiles through API and RDF querying. This approach addresses the issue of interoperability of user profiles generated by heterogenic learning application. However it still requires programmers making decisions at design time about the user profile processing and adaptation rules. The actual users (teachers and students) have only access to the actual data stored in the profiles, they have no access to the rules that build these profiles or which adapt the functionality accordingly.

Open user models in learning communities require users to scrutinize the UM. In online communities not all of the aspects can be handed over to all users. Sometime we need a special user, such as the community owner or administrator, to manage the UM and underplaying rules that create, update and utilize the user model. However there is a need to communicate these rules to general users of the community in some appropriate form, to enable the general membership to have input on how their information is used.

We need a mechanism to request changes that reflect changes in the community's purpose and stage of development, thus adapting also to the individual users needs and purpose.

Self (1999) raised several questions regarding the implementation of open user models, for example "*What part of user model should be viewable by which user*?" One of the questions that has gained relevance recently is '*Should the reasons for the content of the learner model be viewable*?' This question stress the need to visualize the reasoning mechanism of the UM that causes changes in the model. One can raise a follow up question: "*Should the reasoning mechanism that builds and utilizes the model be editable by users?*"

In order to expose the inner workings of the user model to users, designers have to make decision what objects are viewable by which user. This leads to the question, how another designer externalize these decisions, so that in case of any future change, these decisions can be reviewed. This will not be possible if the decisions are implemented in the code. Systems must externalize the storage and reasoning framework for user models and adaptation so that users in special roles (administrators, owners, teachers) can inspect and edit these decisions. By using a standard we can also make these decisions interoperable in collaborating environments.

2.2.4. Purpose-based user modeling

Purpose-based user modeling (Niu 2005) is an implementation of the decentralized, active user modeling approach that involves computing distributed and fragmented user models from various decentralized sources for a specific purpose. The purpose defines the process and the user data types required as input and output. Each purpose either computes a new user model data type and/or provides a certain application-dependent adaptation. Thus, a purpose is an independent processing unit, which can be applied to whatever appropriate user fragment data is available at the moment from distributed and possibly decentralized sources. The purposes can work together in an anytime manner in a hierarchy based on abstraction. More specific purposes positioned towards the leaf nodes are executed when more data from fragmented sources is available, while more general purposes near the root typically demand less data or easier to access data. The purpose-based modeling approach has advantage of providing a local context for computing the model fragment and adaptation. Niu implemented the purpose-based approach in a multi-agent portfolio management system. In this system each agent keeps a user model of other agent based on the transaction and scenario in which the transaction took place (Niu et al., 2004) and makes decisions by integrating the agent's own user model with data about this user collected by asking other agents to send their user model information that is relevant to the purpose and context at hand. With this implementation, Niu, in her thesis (Niu, 2005) provided a "proof of existence" for the decentralized user modeling approach, but did not focus on the important question of how and by whom the purposes will be created. She suggested, as a direction for future work, to focus on creating methods for editing purposes and creating libraries of purposes (just like software libraries) that can be indexed by purpose and context features and searched by agents when the need arises.

Collaborating online communities face similar challenges for user modeling as the distributed, ubiquitous, and agent-based applications. A user who participates in several communities will have models in each of these communities, which will reflect their own adaptation purposes. To achieve richer user modeling and adaptation, it is important to achieve interoperability, updating and synchronization of user models across communities, while preserving their autonomy.

We propose to implement a Purpose-based user modeling approach using policies instead of purposes to compute user models on the fly in online communities. The policy document, similar to a purpose, describes a procedure. A set of policies, like a software library of functions is indexed and searchable, human-readable and editable according to the wishes of the community owner or node administrator. A policy provides all the relevant information for computing a user model and adaptation of the functionality and interface to a given type of user in a given context. This policy based approach allows for a smoother transition of users across communities and enables in this way collaboration of online communities. The proposed approach will be presented in the Chapter 4. The next chapter presents the Comtella community platform in which the approach was implemented and evaluated.

2.3. Summary and Discussion

This survey of different use modeling approaches highlights several shortcomings of existing approaches for user modeling. Currently, in business or other applications the user models are defined at design time according to the business process, functional roles and workflows. However in online social communities the roles, the status, trust relationship of the users evolves as they participate in the community. An agile user modeling framework is required, that allows users (perhaps users in special roles, but nevertheless users, not software designers) to control the user modeling and adaptation process.

However, most of the current user modeling approaches, declarative and procedural, centralized and decentralized, are "black boxes". The information about the user and the rules encoding the decisions of how to make adaptations are hidden from the users. Open user modeling approaches have been proposed to allow visibility and possibility to correct errors in the user models. However no approaches currently provide a way to change the system's rules that create the user model or act upon it. As these rules are defined at design time by the programmers, users have no way to change these rules. We propose a way to involve users in setting the rules for creating and updating user models and the resulting adaptation in the context of online communities. This will result in ever evolving set of rules and adaptation according to context of the online community.

Chapter 3

Comtella: Introduction and Architecture

The policy based user modeling approach requires an online environment where users can create communities and set user policies for their communities. The environment should allow members to participate in more than one community while keeping separate status in each community. To develop such an environment, we decided to use the existing Comtella system for community resource sharing developed in the MADMUC Lab to test reward mechanisms and their impact on the participation of community members.

Comtella was originally designed as a P2P file-sharing application using the Gnutella protocol (Vassileva, J. 2002). It was redesigned in 2004 as a web-based online community that supports students in a class to share URLs related to their topics of study. This version of the Comtella system will be referred in the thesis by "Single Community Comtella" (SCComtella) to distinguish it from the new version "Multi-Community Comtella" (MCComtella), that we developed and which is used as a platform for evaluating the policy based user modeling approach.

3.1 The Single Community Comtella from 2004

3.1.1. Scope and Functionality

The design of SCComtella was aimed at hosting a small community of users for a relatively short period of time to share resources on a fixed set of topics (see Fig. 3.1).

The community had 20 to 30 predefined users who were students registered in a course and used the system to do part of their coursework. Each week the students used SCComtella to share resources (URLs of papers, summaries, and comments) related to the current weekly topic according to the course outline. Thus, the topics addressed by the shared resources were defined in advance by the instructor who served the role of community administrator and not by the users. The current topic changed in time, i.e. each topic was the focus of activities of the whole community for a certain period of time, typically one week. SCComtella was used for two years (spring of 2004 and spring 2005) by two 4th year classes (CMPT 405, “Ethics and Information Technology”) offered at the Computer Science Department of the University of Saskatchewan. Later, when the MADMUC Lab made this software available for use in other classes, the users (class instructors) asked for new features. They asked to have control of the reward mechanism, for interaction between community members. Therefore, the need for new features emerged. Also the idea arose for using Comtella to support sharing research papers among graduate students in a lab for extended period of time, and on a larger scale, across different labs in the department, or different research labs across the world. As the context of the community changed from a single class for a fixed period of time and range of topics, to a more general- purpose community for a longer-term use with unlimited list of possible topics of interest, the software requirements changed significantly.

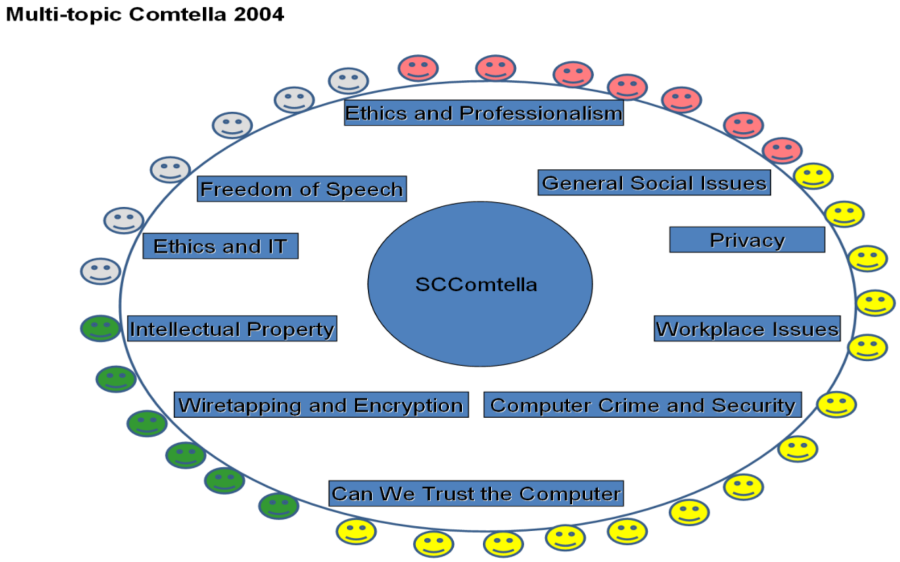

Multi-topic Comtella 2004
Ethics and Professionalism
Freedom of Speech
General Social Issues
Ethics and IT
Privacy
SCComtella
Intellectual Property
Workplace Issues
Wiretapping and Encryption
Computer Crime and Security
Can We Trust the Computer

Figure 3.1 The Single Community Comtella (SCComtella) .
The different colours of the icons representing the users on periphery show their four different possible statuses: "Plastic," Bronze", "Silver" and "Gold"

3.1.2 User Model in SCComtella

In SCComtella the user model represented a certain predefined set of user actions (e.g. share a paper, rate a paper), and two metrics related to these actions – their count and their quality (quality of the paper, quality of the rating) (Fig 3.2). There was also a "community model", which contained the expected number of contributions for the whole class for each topic and a time function which defined how valuable is each type of contributions made at the current moment (e.g. weight for sharing and weight for rating). Based on the community model and individual user model, the adaptation mechanism computed the individual reward factors for every user at the current moment. The points were awarded for different types of user actions, and summed up in a resulting measure reflecting the *user's participation* and stored in the user model. Based on the participation measure, users are classified into four different categories (membership / status levels): "Plastic," Bronze", "Silver" and "Gold", which give different privileges and rights (adaptations of the system’s interface and functionality as shown in Fig 3.3) for the particular user.

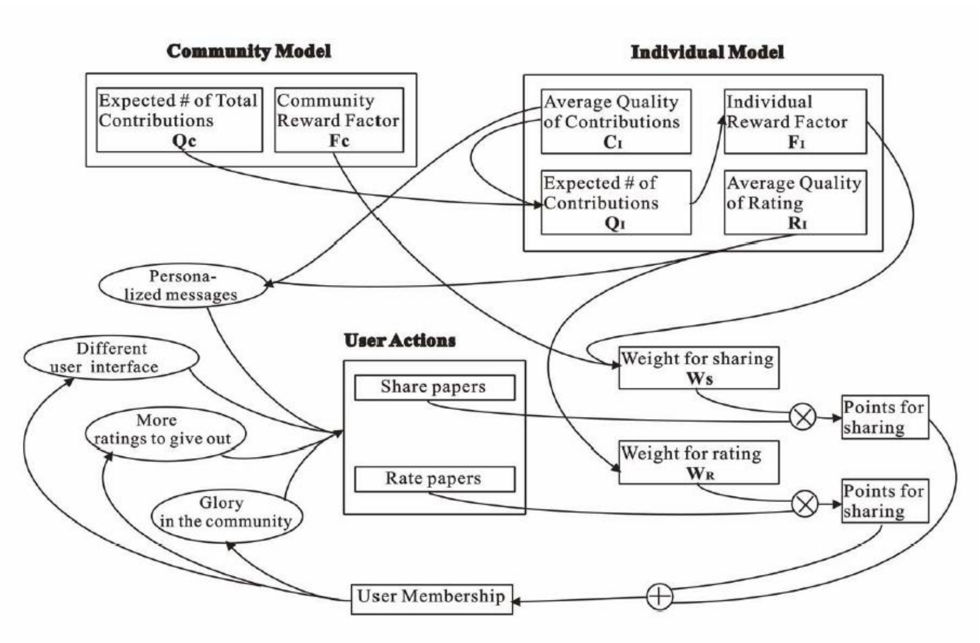
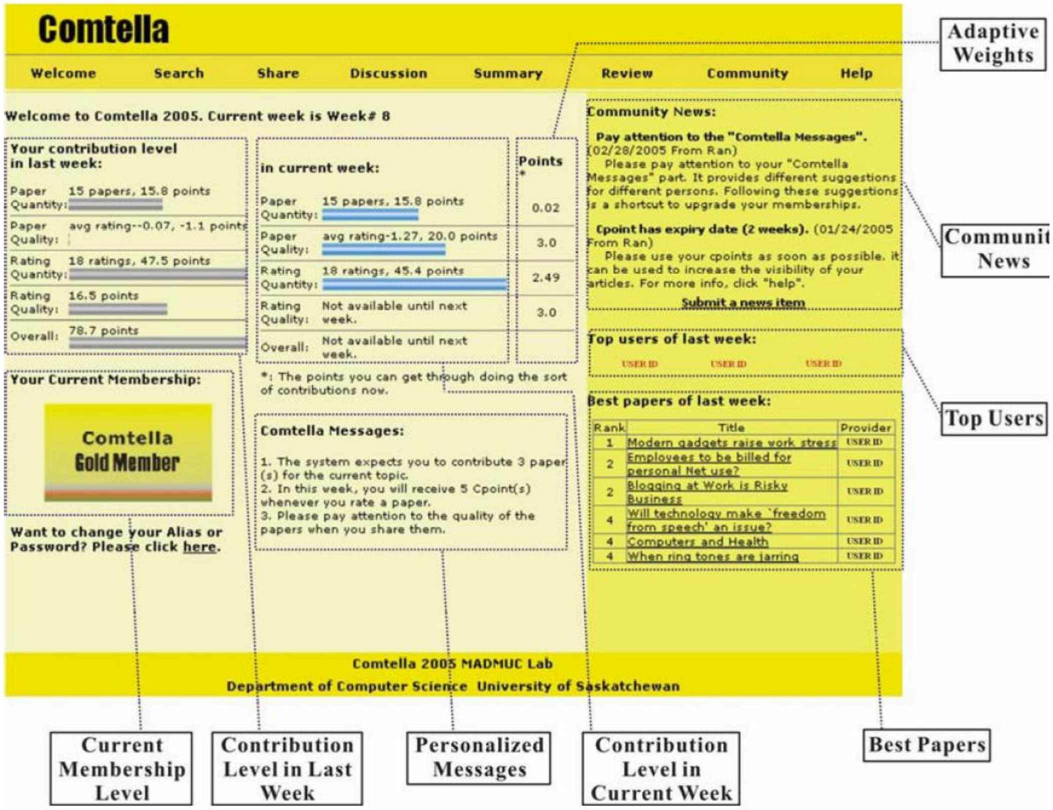


Figure 3.2 Overview of the adaptive motivation mechanism and user model in SCComtella

Reprinted from (Cheng & Vassileva 2005)

This adapted functionality and differentiated rights reflects an embedded motivation and quality control mechanism (Vassileva 2005), “based on theories from social psychology (social comparison, reciprocation)“ which addresses two important issues of online community sustainability: participation and quality of contributions. (Cheng, R. 2005)



Welcome to Comtella 2005. Current week is Week# 8

Your contribution level in last week:

Paper Quantity:	15 papers, 15.8 points
Paper Quality:	avg rating--0.07, -1.1 points
Rating Quantity:	18 ratings, 47.5 points
Rating Quality:	16.5 points
Overall:	78.7 points

in current week:

		Points *
Paper Quantity:	15 papers, 15.8 points	0.02
Paper Quality:	avg rating-1.27, 20.0 points	3.0
Rating Quantity:	18 ratings, 45.4 points	2.49
Rating Quality:	Not available until next week.	3.0
Overall:	Not available until next week.	

*: The points you can get through doing the sort of contributions now.

Your Current Membership:

Want to change your Alias or Password? Please click here.

Comtella Messages:

1. The system expects you to contribute 3 paper(s) for the current topic.
2. In this week, you will receive 5 Cpoint(s) whenever you rate a paper.
3. Please pay attention to the quality of the papers when you share them.

Community News:

Pay attention to the "Comtella Messages". (02/28/2005 From Ran)
Please pay attention to your "Comtella Messages" part. It provides different suggestions for different persons. Following these suggestions is a shortcut to upgrade your memberships.

Cpoint has expiry date (2 weeks). (01/24/2005 From Ran)
Please use your cpoints as soon as possible. it can be used to increase the visibility of your articles. For more info, click "help".

Submit a news item

Top users of last week:

USER ID USER ID USER ID

Best papers of last week:

Rank	Title	Provider
1	Modern gadgets raise work stress	USER ID
2	Employees to be billed for personal Net use?	USER ID
2	Blogging at Work is Risky Business	USER ID
4	Will technology make `freedom from speech' an issue?	USER ID
4	Computers and Health	USER ID
4	When ring tones are jarring	USER ID

Comtella 2005 MADMUC Lab
Department of Computer Science University of Saskatchewan

Figure 3.3 The main page of SCComtella (from Cheng, 2005).

The personalized interface has different colour for users with different status; personalized messages are displayed on the welcome screen, the user gets different numbers of ratings that they can award, and other privileges.

3.1.3 Design

The SCComtella was implemented by Ran Cheng as part of his M.Sc. thesis (Cheng, 2005). It is based on the Model View Controller (MVC) design pattern and allows flexibility to change and add new features.

The MVC model consists of the three components Model, View, and Controller. Model is the actual application logic; a Controller handles requests from users through the View, and updates the Model accordingly. The View is used to interact with users and to provide a window into the state of the model (Burbeck S. 1987). In SCComtella, a Webwork framework is used to implement the MVC. Webwork is web based MVC Framework that separates the View from the Model by using the concept of “actions” (user interaction with the system like search or add link). SCComtella uses a Webwork dispatcher to handle requests and results from user actions. It also uses Struts Tiles to define, organize and present JSP pages as View. Table 3.1 describes the software environments used to build the Comtella online community system.

Table 3.1 MCComtella Software platform

Web Server	Apache Tomcat
Operating system	Windows XP
Programming language	Java
Server side scripting	JSP, Servlet
Web Frame Work	Webwork 2.0
Presentation/layout	Struts Tiles, JSP tag library
Style sheet	Cascading Style Sheet (CSS).
XML Parser	JDOM
Database	MySQL Server, JDBC

An overview of SCComtella architecture and a typical request and response cycle is presented in Fig. 3.4. In Comtella all user requests are received by a main action controller. An action controller dispatches the request to the appropriate action handlers. An action handler implements the actual logic of actions. The Action handlers select required Data Access Object (DAO). DAO implements the logic to work with the

Database. Normally one DAO manages one table in the Database. DAO offers operations such as select, insert, update and delete in a table. DAO also updates the Value Beans to store information for or from a table. Fields in Value Bean objects have one to one mapping with fields of a record. For example, if a user sends a search request with a keyword, the request will be received by the main action controller which selects the Search action handler. The Search action handler will select sharedInfoDAO to do the actual search in the database and update a collection of SharedFileInfo value beans. These Value Beans are then available to view through a Webwork framework tag library for presentation on the web page.

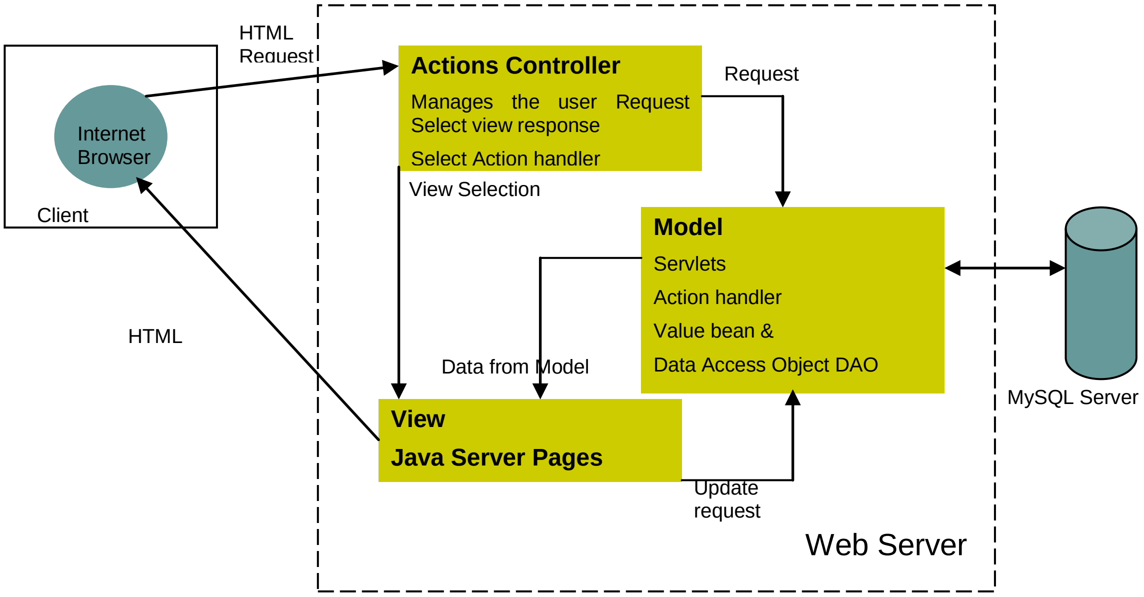


Figure 3.4 A typical interaction in Comtella

3.1.4 Limitations of SCComtella

SCComtella had limitations both for community members and community administrators. Some of the limitations for community members were:

1) There was no process and user interface allowing registration / signup to add new users; users were added directly by the community designer (programmer)

2) There was no process and user interface to add new topics after starting a community; the list of topics was pre-programmed in the database.

3) It was impossible for the instructor to change the reward mechanism or change the interface adaptation criteria. This required a major reprogramming of the underlying mechanisms.

4) It was impossible for users to edit and/or delete already shared resource (links or comments), to upload files, or to logout.

5) There was no communication tool for the community members other than a shared discussion board.

6) There was no mechanism for two communities (e.g. students from two different classes, each using SCComtella) to collaborate, as those were hosted by two separate web servers.

7) SCComtella lacked any community administration tools for adding/deleting members, or sending messages to community members.

In fact, the community administrator in SCComtella was the programmer/designer (Ran Cheng), who had to install and administer a web server and deploy the system on it. Learning the intricacies of the software enough to be able to set up a new system for a given community required approximately 2-3 months.

The SC Comtella (MCComtella) 2004 is focused on supporting a group with a converging interest (e.g. the subject of a class) for fixed period of time (e.g. a term). It can not support well an open community (where new members can join in) in growing set of areas of interest (where new topics can be added at any time) and which persist over time, because it does not have strong annotation and search functionality. The shared space would quickly become cluttered with diverse materials that can not be found easily using just the limited topic-based menu. Also, it cannot support new groupings, for example, an institutionally- or geographically -based group, or a group with particular area of interest cannot be added.

To support open, long-term communities, with specific areas of interests, the SCComtella was redesigned and a new version was created as part of this thesis work. The resulting Multi Community Comtella (MCComtella) is described in the next section.

To support geographically or organizationally based communities, the MCComtella was further extended to allow for different "nodes" hosting community clusters. The resulting Multi-Node Multi-Community Comtella (MNMCComtella) is described in section 3.3.

3.2 Multi Community Comtella

3.2.1 Functionality

The new Multi Community Comtella (MCComtella) has the following new functionality in comparison to the previous SCComtella.

1. **New User Registration Process:** To support openness (i.e. the possibility of new users joining the system), a new user registration process and form is added which allows users to sign up for MCComtella. This sign-up form provides the user with a virtual identity that can be used to join communities. The SCComtella did not allow users to register themselves; new users had to be entered in the database by a programmer.

2. **Support of Multiple Communities:** Users can create their own communities. The user who creates a community becomes its owner and is responsible for the administration of the community. The owners can add keywords describing their community to help users find this community among the others. In contrast, the SCComtella had just one community, but various predefined topics for which users could share resources by selecting a topic appropriate for the shared resource from a menu.

3. **Private and Public Communities:** The communities in MCComtella can be either private (other users can join then only per owner's permission) or public (open for anyone). MCComtella users can participate in any public community and becomes a member of it. For private communities, they have to send a request

to the community owner to join. The community owner can accept or decline their request. In contrast, the community in SCComtella was private, limited to the registered users only.

4. **User Management:** A community owner can remove any member from a private community. MCComtella offers community administration tools which allow actions such adding or deleting members in private communities. In SCComtella, user management was done directly in the database by a programmer.

5. **Multiple User Profiles:** Users can have different status (membership) in different communities depending on their contribution in a respective community. As users participate in a given community, their status in this community is updated instantly on each contribution or action according to certain rules (policies) set up by the community owner. In SCComtella users had one status (membership) only, since they were all members of the same community.

6. **Default Community:** Users can choose a community as their default community. The status of user and other information regarding this default community is displayed on the welcome screen the user sees first when they log in. After that the users can participate in their default community or switch to any other community.

7. **Tag based Search:** To facilitate search, users are strongly encouraged to tag the shared resources. The tags used for all shared resources by a community are listed on the community page (tag cloud) and users can view the links/files annotated with a given tag by clicking on this tag. In contrast, in SCComtella resources were

shared for particular topic selected from a menu. Users were able to view the shared resources by selecting the topic from the menu (usually a list of results that was several pages long).

8. **Global Search:** Users can search for resources by clicking on a tag in the list of tags for their current community. In addition to this they can also search across all communities hosted in MCComtela, by using the “Search box”. In this way they can discover communities they are not aware of. There is no equivalent functionality in the SCComtella.

9. **Manipulating Resources:** Users can share links, upload files, tag, rate, and comment on the already shared resources. They can also delete and/or edit the resources (links, or files) that they have contributed themselves. They can add keywords while sharing a link/ file. In contrast, in SCComtella users could not upload files, and could not delete and edit their own contributed resources.

10. **Favourite Communities and Resources:** Users can add any number of communities in "My Communities" to make a list of favourite communities. User can also manage have a list of favourite links.

11. **Highlighting New Resources:** In the search screen new links (resources added after the user’s previous session) and files are marked with a different color and font.

12. **Favourite Community Members:** Users can set a particular community member as his favourite. Similar to subscribing to a feed, the user will get notified when their favourite community member adds a new link or file.

All these functionality extensions made MCComtella a powerful open community framework allowing for new groups (communities) to be started by regular users at any time. It gave more power for users to manipulate their resources, to annotate them, and to search for resources both in their current community and across all communities.

3.2.2 MCComtella Design

In MCComtella any user can create a new community and can participate in more than one community maintaining a separate profile in each. These new features required implementation of three new frameworks in MCComtella: a user management framework, a framework to support multiple communities and a policy based framework for user modeling in these communities (see Fig. 3.5). The session manager works as proxy for user to interact with the different frameworks (see Fig 3.6). For user register and login it works with the user management framework, to allow users to create and join a community - with the multi-community framework and to allow user to develop and read a policy of the community - with the policy framework.

The user management framework consists of modules to register new users, manage user status and roles in any given community. The multi-community framework contains modules to manage community ownerships, community membership, and the metadata of communities such as description, tags and community search interface. The policy framework contains modules to manage the status policies and the role policies in each community. The policy framework is used by user management framework to update user status and roles, as will be explained in the next chapter. The multi-community framework works with user management framework to ensure community administration.

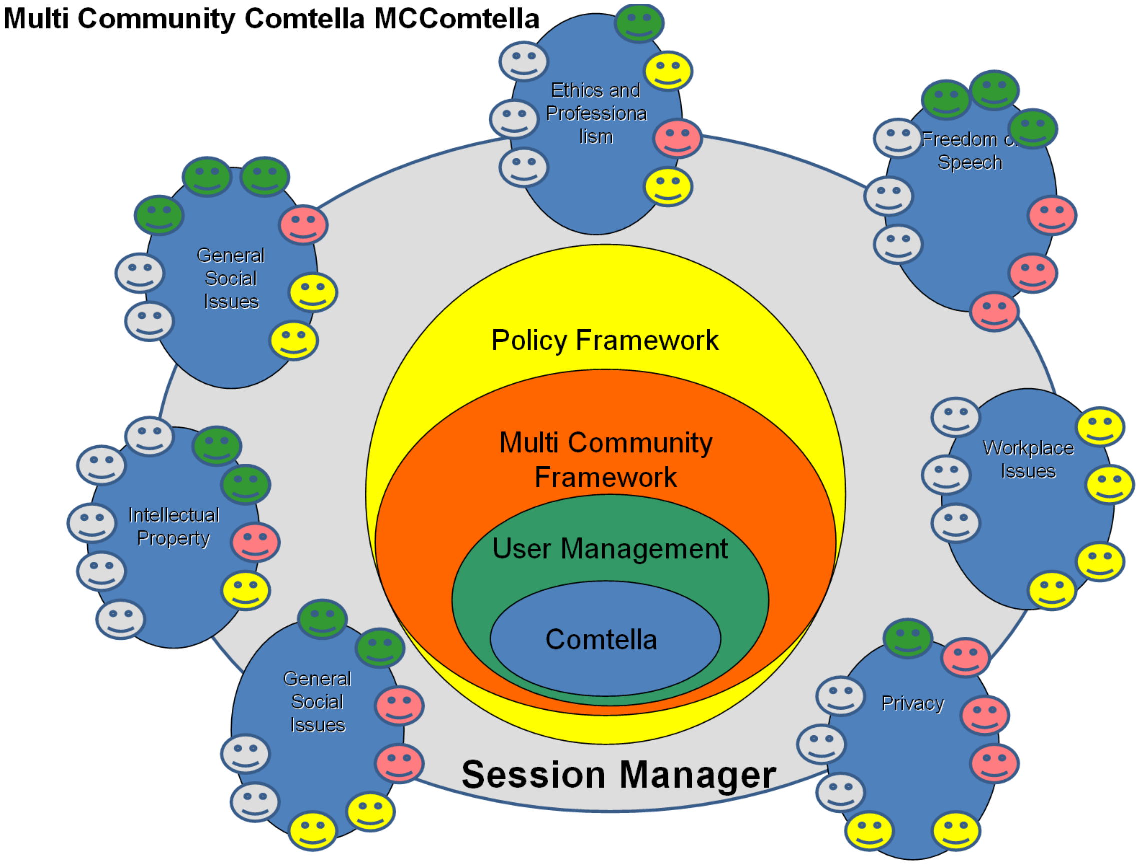
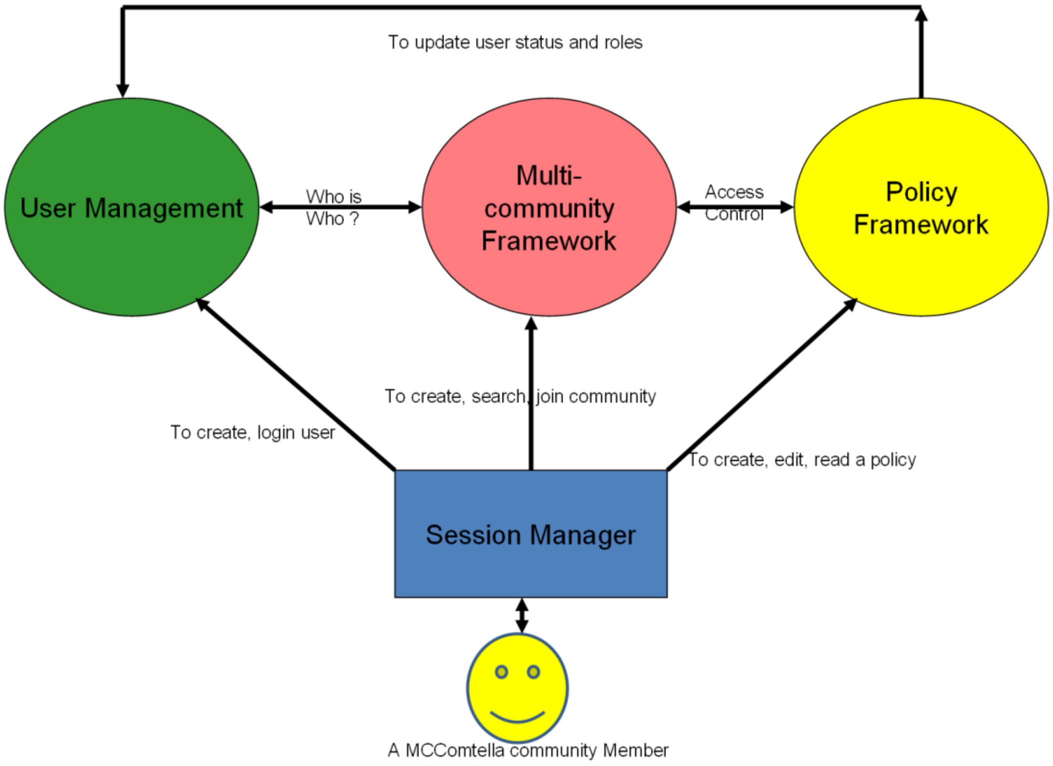


Figure 3.5 Multi Community Comtella (MCComtella)

The figure shows multiple communities focusing on different topics of interest, each has users with different status. The communities are managed by three frameworks overlaying the Comtella kernel and coordinated by the Session Manager.



To update user status and roles
User Management
Who is Who ?
Multi-community Framework
Access Control
Policy Framework
To create, search, join community
To create, login user
To create, edit, read a policy
Session Manager
A MCComtella community Member

Figure 3.6 Frameworks for user, multi community and policy management in MCComtella

These changes have resulted in a large complex system. It has 33,292 lines of java code in 161 classes and 126 Java server page files. It handles 92 different actions from users in the interface. For these actions it uses 29 database access classes to interact with the MySQL database server. These database access classes populate 40 java beans. The system design, however, minimizes this complexity by using software engineering principles and design patterns, such as MVC. The Session Manger decouples the various frameworks and their modules, which makes updates easier.

3.3 Multi-community and Multi-Node Comtella

An observation made by Mynatt et al. (1997) suggests that online spaces take their conventions and norms from physical spaces. Users extend their relations from the real world into the virtual world. Similar geographic and demographic features of participants reinforce the bond between people in the virtual world. The physical space has an impact on the participation trends as well. People sharing the same geographical, organizational, or cultural background will more likely interact with each other in the virtual world (Indratmo & Vassileva 2005). This phenomenon can be observed on Internet Relay Chat (IRC) where chat channels that are named by a city, or language have large participation. At the same time, people feel more comfortable in forums hosted by some local authority like school, university or organization. Therefore, it would be beneficial to let the communities grow within an organizational web server (called"node"). However, it is important to provide a framework for collaboration and interlinking of these distributed small clusters of online communities.

For this reason, the design of MCComtella was extended with a multi-node architecture, allowing communities to be hosted in different websites (nodes). For example imagine two schools hosting two different MCComtella nodes, (http://kardam.usask.ca) and (http://karachi.usask.ca) on two different web servers. Any node can host multiple communities, for examples, for their classes in math, social science, chemistry or photography. These communities can collaborate within and across nodes, i.e. students who participate in the social science community can visit the math community in their own school, at the same MCComtella node, and also the social science or math community in the other school on the other Comtella node. The main differences of the new design of Multi node Comtella are summarized below:

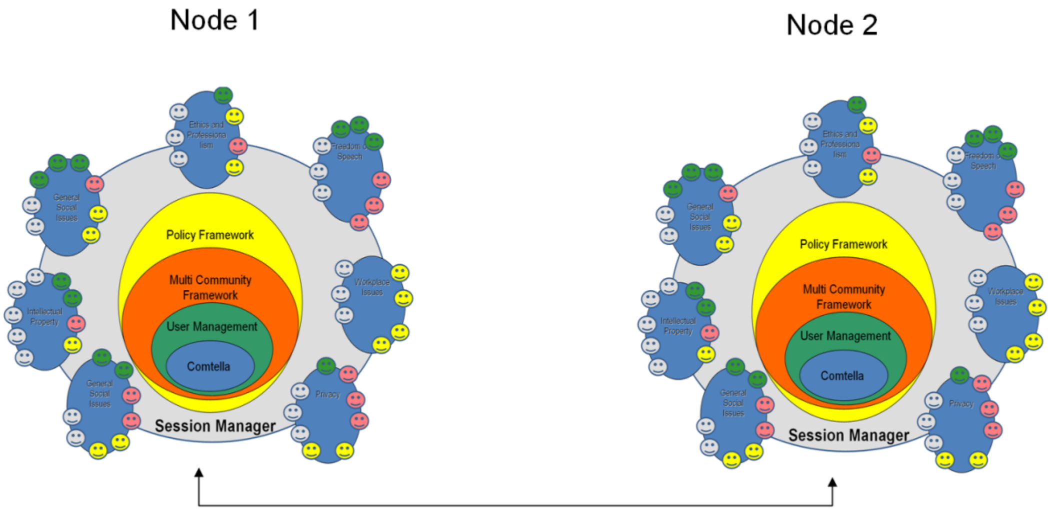
- The communities may be hosted at one web server (node), (as in the previous Comtella design) or they may be hosted by distributed nodes.

- Each node serves the purpose and interest of the local hosting organization. For example, one scenario can be the commissioning of Comtella server in different high schools (nodes), each hosting several communities, e.g. one for each subject with a teacher as a creator. Each community is autonomous and may have different policies for rewarding participation.

- The communities hosted by one node can interact locally and across the nodes, where "interaction" refers to the possibility for users who typically participate in one community, to search, find, read, comment, and rate resources shared in another community. Interaction can happen across nodes too. For example, two class-based communities for the same subject on different nodes (school) can establish an interaction.

To implement the communication between nodes the design extends the functionality of the session manager (see Figure 3.7). The session manager identifies each user from the default community. A user who comes from a community that is not local to the current node, it will redirect the user to his home node to fetch the use credential and policies. Session manager establish the validity of user by using these credential and local policy.

Multi Community Multi Node Comtella



A session manager can identify the users from another node and use the policy framework and multi-community framework of respective node acquire members info. and transfer user from node 1 to node 2

Figure 3.7 Multi-community and Multi-Node Comtella:

Each node is separate Comtella website. Session managers on each site are responsible for user transfer**.**

3.4 Comtella in Use

SCComtella has been in use with two 4th year classes in 2004/2005 - on “Ethics and IT” and “Multi-Agent Systems”, each with approximately 20 students for the duration of one term. It was also used for the Adaptive Web class in 2005-2006 in Department of Information Science and Telecommunications School of Information Sciences University of Pittsburgh. The MCComtella was offered as a resource-sharing tool to the User Modeling (UM) research community at the biannual conference in June 2007, pre-filled with all the materials shared in the Adaptive Web class. However, it did not get much use from researchers from the UM community apart from some occasional sharing of photos and testing. The system is currently used by the NSERC Chairs for Women in Science and Engineering and their staff to share resources (meeting notes, resource papers and education materials, reports, proposals and statistics about Women in Science and Engineering). This system is available at http://comtellawise.usask.ca (the access is limited only to assigned users).

3.5 Why is MCComtella a Good Testbed for a Policy Framework?

The new functionality in MCComtella implicates new requirements that have to be considered in the modified design:

- Each community is autonomous, managing its own local rewards and status reward mechanisms. Community owners need a mechanism to communicate these local policies to new users.
- Communities can change their reward mechanism any time. Community owners need tools to make these changes, so the community owners need to set or change their local policies. User policies such as reward mechanisms, access rights and roles should be decoupled from the application logic and editable.
- Allowing users to interact with multiple communities implies that they will have different level of interest and involvement in them. This will lead to users reaching different level of membership (e.g. gold, silver, bronze) and taking different roles (e.g. a community owner or a regular member) in different communities. Interpretation of the value of different level of memberships across communities and

the corresponding adaptation (rights and privileges) can only be decided by the community owners themselves.

These needs of MCComtella (and also Multi-Node Multi-Community Comtella) match well with the features that we are looking for in online communities suitable to implement our user modeling approach:

- Many online communities are not open source, and usually both user and usage data are proprietary, therefore not available for any independent study. MCComtella is a system developed in-house, it is based on open source technologies; so it offers user (with privacy protected) and usage data for research purposes.
- We need to have several online communities that share user identity and user model features/attributes. In this way we avoid dealing with the problems of ensuring a common user identity necessary to be able to follow the movement of users across communities. We also avoid dealing with and mapping across different ontologies for user modeling. Both of these problems are important problems that need to be solved to achieve practical interoperation between online communities. However, currently all online communities have their own user identity management and user attributes, though there have been some attempts to develop standards, e.g. OpenSocial. Adopting standards is a slow process and it wouldn’t be wise to wait. MCComtella uses a common set of user attributes (ontology) to model users in all communities and a shared (centralized) user identity across all communities and nodes, so it is an ideal implementation test-bed for the policy-based user modeling approach.

Chapter 4

An Open Policy Based Approach for User Modeling and Adaptation in Online Communities

4.1 What are Policies?

A colloquial definition of the word “policy” from Wikipedia is as follows:

> *A **policy** is typically described as a deliberate plan of action to guide decisions and achieve rational outcome(s). However, the term may also be used to denote what is actually done, even though it is unplanned.*

Policies are used widely in the information technology domain. The most common application of policies is to set security and access rights for users in a given system (Lorch et al. 2003). For our purpose, the policies are rules and constraints governing the user interaction and resulting adaptation of the system, to ensure that all users can fulfill their purpose. Policies are applied to ensure security, consistency, and fair availability of recourses and services.

Policies are normally hidden from the users and are focused on protection and privacy of resources in a computer system (Anderson 2006). However, in online communities users play an active role in contributing and managing content and it is important that they are aware of the existing policies to be able to behave responsibly. It is important also that users participate in the creation of policies. We need to protect the policies from unauthorized change, but also publicly display them for community members so that they know the rules and the consequences of their actions. In current implementations policies are typically deployed to define user roles and respective privileges. In the proposed approach policies are used to define various classes of users, classify users into these classes and specify functionality and interface adaptations suitable for these users. Thus in fact policies implement both an adaptation mechanism and user-modeling mechanism.

XACML is an OASIS standard that describes both a policy language and an access control decision request/response language (both written in XML). The policy language is used to describe general access control requirements, and has standard extension points for defining new functions, data types, combining logic, etc. The request/response language allows to form a query to ask whether or not a given action should be allowed, and interpret the result. The response always includes an answer about whether the request should be allowed using one of four values: Permit, Deny, Indeterminate (an error occurred or some required value was missing, so a decision cannot be made) or Not Applicable (the request can't be answered by this service).

4.2 User Policies in Online Communities

When designing an online community the complexity of user policies should be addressed at two levels. First, user policies should be maintainable in response to changes within the community which can occur in the course of time, and second, user policies should be negotiable at user interface level to allow community owners to transfer user policies when collaborating with other communities.

4.2.1 Policies within a community

Users in the scope of a single community can be different from each other due to differences in their roles, implying different access rights and relevant policies. According to user entitlements, the system grants or denies access to different resources. Currently, online community designers are implementing the user policies implicitly in the code, by storing data from the user models in databases, while an application or program makes decision about user authorization and access rights directly using the user model data. This simple method of user identification can not handle the complexities of online communities, where rights of access, rating and commenting can vary according to the role, timing, user reputation or status and relationships.

The needs for adaptation in online communities change in time and this demands frequent changes in user policies. For instance, at an early stage for the life cycle, a community owner or moderator may prefer to reward higher quantity of contribution than quality, to achieve faster a critical mass of participation, but at a later stage, they may prefer to reward only high quality contributions (Cheng and Vassileva 2005). Any such change in policies will have ripple effects across the architecture of the community if it is designed with the current database-centric approach. Therefore the user model and the user policies should be represented explicitly, decoupled from the code and the other components to minimize the impact of change. The representation should be explicit enough so that a user is able to understand it. If a community owner wants to change the relative weight or wishes to ignore one or more user activities or introduce a new activity update the user model, they should be able to view the current policy, understand its implications and the implications of a change in the policy. The community owner should have a fine grain control on the customization of user policies, to reflect the true needs and sustainability of community.

4.2.2 Policies across communities

Although online communities attract people with their services and interaction tools, the long term sustainability of communities depends on the social relationships developed among the members. These relationships can be either between individuals or among people in groups. Providing tools to create and maintain such social relationships becomes an important concern with the increasing cross community interaction (Cold, 2006) resulting from technologies such as RSS and web-services. Service-based architectures, while providing a loose coupling between websites for collaboration, require open user policies. Both service provider and consumer communities need a negotiable or at least an inter-operable user policy. RSS and mashup technologies allow deep interlinking across communities. This interlinking, while providing a solution of intercommunity collaboration, also raises the question which user policies should apply to users across community “borders”. Some of the issues, resulting from across community interactions are as follows:

Authentication: Web services and other collaboration techniques bypass the conventional “sign in” pages. This raises the question of how users would be authenticated on the service provider site.

User Identity: User identity is a unique set of attributes that includes him/her in a group and distinguishes within the group. Many questions arise about the user information. For instance, if a service provider site needs user information to provide a service, which information should be carried across? How this information will be interpreted in the new context? How users can trust the new users from a different community.

User Model: In case of online communities it is not just a problem of access control on certain services. Users have certain roles and reputation that is well known to the other members within a community. These roles and reputation are based on community-specific policies Therefore, in case of cross community interactions; there must be a way to communicate the role and reputation of the newcomer in machine readable format

User Policies: Many questions arise, pertaining to user policies. For instance, how conflict among policies of different communities would be resolved? Two solutions can be: a *federation* for identification and verification of polices, or *negotiation* among communities about the policies that would apply for each specific case “on the fly.” Another interesting question is how collaborating communities will handle change in user policies.

Let us take as an example an air traveler who has a Gold Membership Card of Safari Air which entitles him certain privileges and access to the executive lounge of that airline. Imagine that he lands at an airport, where Safari Air does not have any facility to provide services to its members. However, Safari Air has a contract (or a policy) with another airline, Blue Air to provide services to its Gold members. Blue Air has its own system of five membership levels (expressed in its own customer reward policy) that include Troposphere level, Stratosphere level, Mesosphere level, Thermo, and Exo-sphere levels. In order to provide services to this traveler airline need to verify that he is a valid Gold Member of Safari Air. This would require a shared mechanism of authentication between the two airlines. They need to share information of their internal reward policies describing the possible customer membership levels, how much miles are required for each level and their respective privileges. Each of the companies will also need to create their own "transfer" policy describing how to treat passengers who have a certain level of membership in the other company, i.e. create a mapping of the customer memberships of Safari to those of Blue Air. For example, a Gold member in Safari Airlines may be treated as a Stratosphere-level passenger and given access to the Blue Air Stratosphere lounge.

4.3 User Policies in MCComtella

MCComtella allows users to create communities. According to a special policy, every user who performs the action "create a community" is assigned the role "Owner" of this community. This role is stored in the user model and the adaptation corresponding to users who have this role unlocks special functionality which allows them to set different kinds of policies for their community: *"Status Policies, "Role Policies", "Access Policies" and "Transfer policies"*.

For example, the Status Reward Policy is a kind of Status Policy. This policy describes the weight of different actions in computing the participation score for the user. The set of rewarded actions can be selected from a list of identifiable actions in Comtella, and may include *Upload a File, Share Link, Rate a Link, Post a Message/Comment, Create a New Community*. The selection of the actions and rewards depends on the community objectives. For example the community owner wants to encourage more discussion, the *"Post Message"* action may be given the highest reward. If the focus of the community is on high quality contributions, then the *"Rating"* action should have a relatively high reward. Therefore according to the objectives of community, the user who has created the community and has the role of community o*wner* can reward and or ignore certain user actions.

When a new user signs up in MCComtella, she is assigned to a default community. The user can search or browse the existing communities and choose to join another one and make it her default community. MCComtella offers a "*Community Search*" option for user to browse through the available communities in MCComtella. A user's status is determined through the *"Status Policy"* in her default community. In every other community the users are assigned temporary status according to this community's "*Transfer Policy".*

In MCComtella the user models are created and updated independently in each community according to the policies of this community. When users switch their default communities fragmented user profiles arise in all of the previous default communities. To ensure interoperability of these profiles, a trust relationship among the communities is necessary. Such relationship can be established between the owners of two communities and is materialized as a set of transfer policies between the two communities. The transfer policies command the transfer of user data along with the user's identity (all communities in MCComtella has shared user identity format) to any new community where a new user model can be established according to the context.

Access control policies implement rules and conditions under which users can perform actions on a resource, such as reading, rating, replying, commenting, or deleting a posting. Usually access control policies are the basic policies that are used by higher level policies, such as status-, role- and transfer-policies to express specific decisions, e.g. allowing or disallowing a user request.

4.4 Why Do the Policies Implement Open User Modeling Approach?

The policy based approach opens the user modeling and adaptation mechanism to the users. It externalizes the user actions that are fed into the user model (e.g. share a file, rate a link, comment on a paper, etc.), and the measure of their value (weights) so that they are visible and editable. In SCComtella these user actions and weights were fixed and developed by programmer at the start of community (so it implemented a "black box" user model and modeling process). In MCComtella, in contrast, the model is open (both the data inputs, and the mechanism) to the user who is in the role of community owner and it can be modified at any stage of the community's life (see Fig 4.1). Policy based approach separate the User Modeling Process from the Adaptation Process. The Community Owner has input on parts of the user modeling process (the actions, the weights, the thresholds /levels). This influences how the User Model is built (the user model here is the User membership/status). On the other side, based on the User Model (membership/status), the Adaptation Process takes place. The Community Owner has input on the Adaptation Process by changing the adaptation policy.

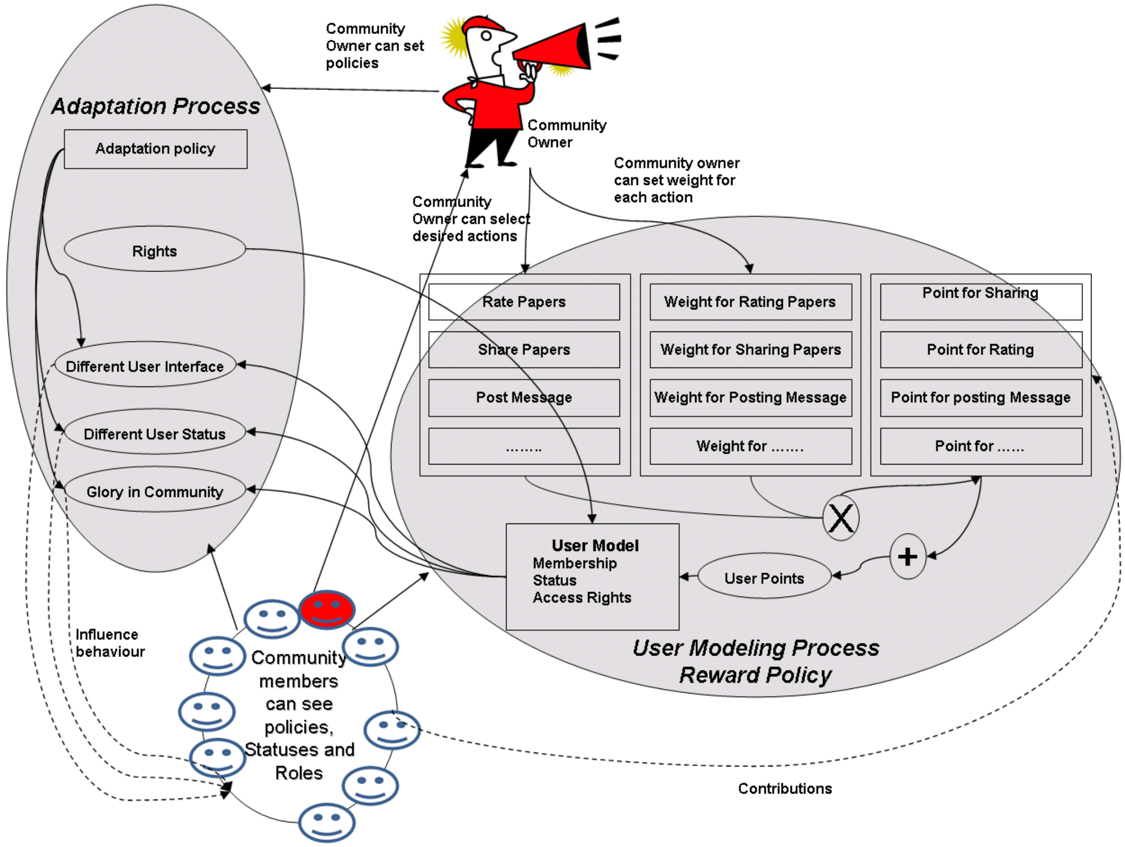


Figure 4.1 Overview of human-centered policy based user model in MCComtella

MCComtella encodes these actions, weights and resulting adaptation in XML documents, called “*policies*”.

To illustrate why the policies implement open learner modeling let’s discuss in more detail the status policies which update one particular feature in the user model – “status/membership” – and the user permissions and adaptations that are to be carried out for users with different status/membership.

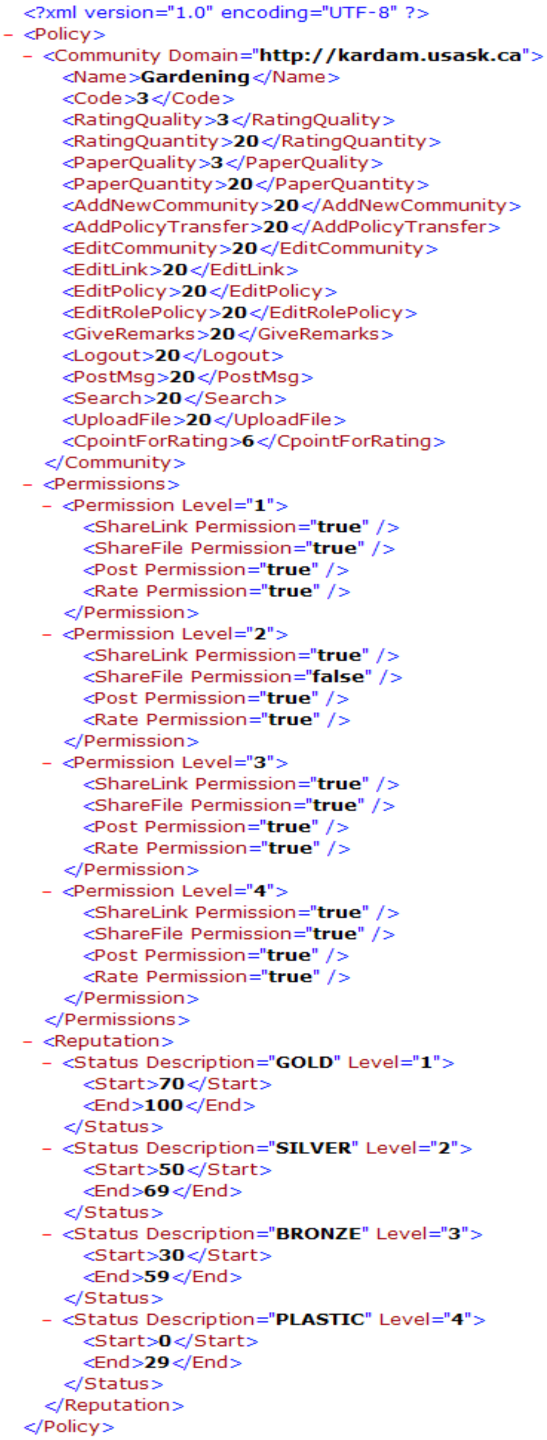
1) The ‘Status Rewards Policy’ – defines rewards for participation listing all desired actions and their respective weights. This policy, in fact, expresses what input data from user actions should be considered in the model and how it should be interpreted to generate a feature (the user participation score) represented in the user model.

2) The ‘Status Update Policy’ – updates the status level based on participation score. It assigns the thresholds for participations points that define each status/membership level. This policy, in fact, expresses how data is processed inside the user model. Both the participation score of the user and the user status/membership level are features of the user model; the policy defines the “conversion” from participation points into status level.

3) The ‘Status Permission Policy’ – defines the status based access permissions for different actions for users with different given status levels.

4) The ‘Status Adaptation Policy’ -- defines the status based interface adaptation and privileges for each status/membership level. This policy matches a particular set of features of the user model (in this case the user status) to specific adaptations (features / functionalities in the user interface).

Fig 4.2 shows these three types of status policies for a particular community in MCComtella – the “Gardening Community”.



```
  <?xml version="1.0" encoding="UTF-8" ?>
- <Policy>
  - <Community Domain="http://kardam.usask.ca">
      <Name>Gardening</Name>
      <Code>3</Code>
      <RatingQuality>3</RatingQuality>
      <RatingQuantity>20</RatingQuantity>
      <PaperQuality>3</PaperQuality>
      <PaperQuantity>20</PaperQuantity>
      <AddNewCommunity>20</AddNewCommunity>
      <AddPolicyTransfer>20</AddPolicyTransfer>
      <EditCommunity>20</EditCommunity>
      <EditLink>20</EditLink>
      <EditPolicy>20</EditPolicy>
      <EditRolePolicy>20</EditRolePolicy>
      <GiveRemarks>20</GiveRemarks>
      <Logout>20</Logout>
      <PostMsg>20</PostMsg>
      <Search>20</Search>
      <UploadFile>20</UploadFile>
      <CpointForRating>6</CpointForRating>
    </Community>
  - <Permissions>
    - <Permission Level="1">
        <ShareLink Permission="true" />
        <ShareFile Permission="true" />
        <Post Permission="true" />
        <Rate Permission="true" />
      </Permission>
    - <Permission Level="2">
        <ShareLink Permission="true" />
        <ShareFile Permission="false" />
        <Post Permission="true" />
        <Rate Permission="true" />
      </Permission>
    - <Permission Level="3">
        <ShareLink Permission="true" />
        <ShareFile Permission="true" />
        <Post Permission="true" />
        <Rate Permission="true" />
      </Permission>
    - <Permission Level="4">
        <ShareLink Permission="true" />
        <ShareFile Permission="true" />
        <Post Permission="true" />
        <Rate Permission="true" />
      </Permission>
    </Permissions>
  - <Reputation>
    - <Status Description="GOLD" Level="1">
        <Start>70</Start>
        <End>100</End>
      </Status>
    - <Status Description="SILVER" Level="2">
        <Start>50</Start>
        <End>69</End>
      </Status>
    - <Status Description="BRONZE" Level="3">
        <Start>30</Start>
        <End>59</End>
      </Status>
    - <Status Description="PLASTIC" Level="4">
        <Start>0</Start>
        <End>29</End>
      </Status>
    </Reputation>
  </Policy>
```

Figure 4.2 XML document to store the status policies for a given community

In the next section the different types of policies are discussed.

4.5 Different Types of Policies

4.5.1 Status Policies in MCComtella

The objective of these policies in MCComtella is to motivate user participation in a community by providing special privileges to users who contribute to the community (Chang and Vassileva 2005). The status update policy defines the user participation metric (see Table 4.1) – what user actions are taken into account and how they are rewarded. This policy specifies how the individual user model in Comtella is updated. The status update policy defines the threshold values required for the users' participation measure to acquire a given status (see Table 4.2). The status adaptation policies define the adaptations and permissions for users with particular status level. A status policy defines the associated access rights (Table 4.3), another the adaptations in the interface and functionality that will be made for a user with a certain status level (Table 4.4) . For example the gold status users in Comtella have access to the gold-coloured interface frame, while plastic status users have access to a green-coloured interface (see Fig. 4.3). In addition, higher-status users in MCComtella receive more ratings to give out, which indirectly affect the available functionality to them. Thus the status policies in fact define the entire user modeling and adaptation loop: from the data to be collected, through the way it is processed, to how the user model is updated and what adaptation decisions are taken based on the model. The procedure of computing values can be one of a library of possible procedures to be chosen by the owner (even though in current implementation, it is fixed).

Table 4.1 Status Policy – rewards for participation

Policy to update user status		Description
Policy Type:	Status	This is a policy to control the parameters of status calculation at Node http://kardam.usask.ca . MCComtella rewards user for some desired activities which includes Rating Paper and Sharing paper. Community owner can control this reward by changing the parameters like weight for the frequency and quality of the activity. By rating a paper user gets Cpoints which he can use to get prominent position for his shared papers.
Effective Date	Jan 10, 2007	
Node	http://kardam.usask.ca	
Community id:	1	
Community Title:	Pictures	
Weight for Rating Quality	4	
Weight for Rating Quantity	3	
Weight for Paper Quality	4	
Weight for Paper Quantity	3	
Cpoints for Rating	6	

Table 4.2 Status Policy – status level based on participation

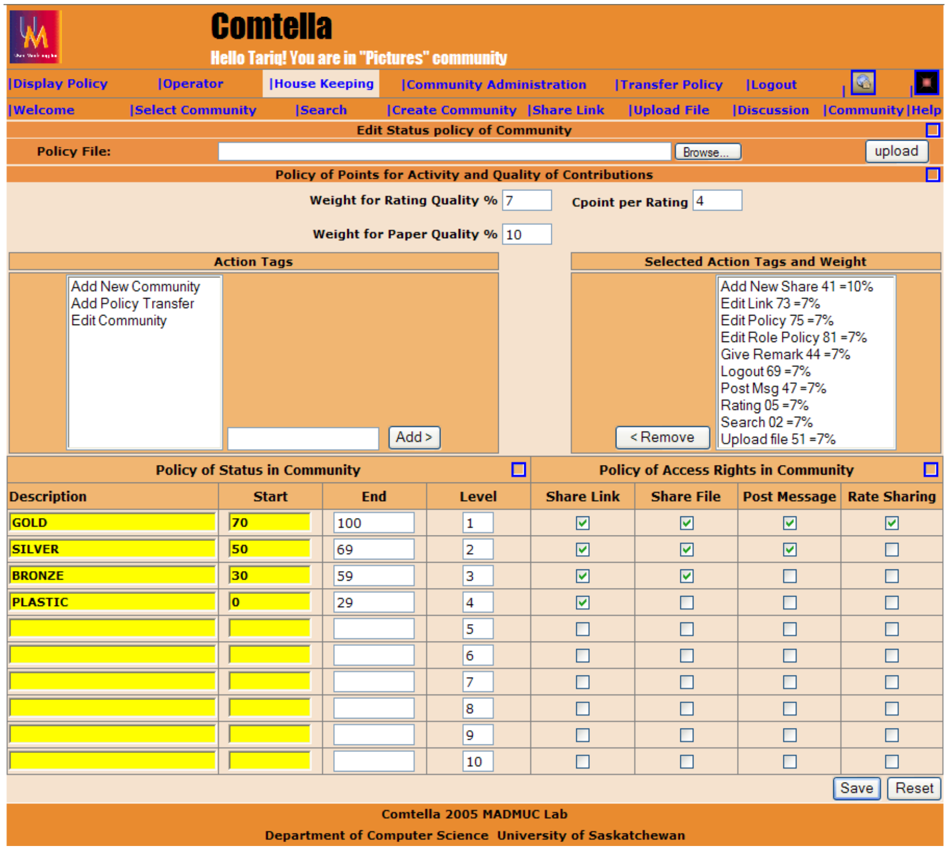
Status level definition policy				Description
Level	Description	Start Value	End Value	User can Name the range of points earned through status policy into status level. For example Gold, Silver, Bronze and Start.
1	Gold	700	1000	
2	Silver	500	699	
3	Bronze	300	499	
4	Start	0	299	
	...			

Table 4.3 Status Policy – status based permissions

Status Permissions policy √ Action allowed × Action not allowed					Description	
Level	Description	Share link	Share File	Post	Rate	Based Status level community owner can set permission for each status level. These permissions are used for interface adaptation and access rights on actions
1	Gold	√	√	√	√	
2	Silver	√	√	√	√	
3	Bronze	√	√	√	√	
4	Start	√	×	√	√	
	....					

Table 4.4 Status Policy for status based interface adaptation and privileges

Status	Interface colour	Rating to give
Gold	Golden	50
Silver	Gray	30
Bronze	Reddish	20
Start	Green	10
....		



Edit Status policy of Community

Policy File:

Policy of Points for Activity and Quality of Contributions

Weight for Rating Quality % 7

Cpoint per Rating 4

Weight for Paper Quality % 10

Action Tags

Add New Community
Add Policy Transfer
Edit Community

Selected Action Tags and Weight

Add New Share 41 =10%
Edit Link 73 =7%
Edit Policy 75 =7%
Edit Role Policy 81 =7%
Give Remark 44 =7%
Logout 69 =7%
Post Msg 47 =7%
Rating 05 =7%
Search 02 =7%
Upload file 51 =7%

Policy of Status in Community				Policy of Access Rights in Community			
Description	Start	End	Level	Share Link	Share File	Post Message	Rate Sharing
GOLD	70	100	1	☑	☑	☑	☑
SILVER	50	69	2	☑	☑	☑	□
BRONZE	30	59	3	☑	☑	□	□
PLASTIC	0	29	4	☑	□	□	□
			5	□	□	□	□
			6	□	□	□	□
			7	□	□	□	□
			8	□	□	□	□
			9	□	□	□	□
			10	□	□	□	□

Comtella 2005 MADMUC Lab
Department of Computer Science University of Saskatchewan

Figure 4.3: Interface for Editing a Status Policy in MCComtella

4.5.2. Role Based Policies in MCComtella

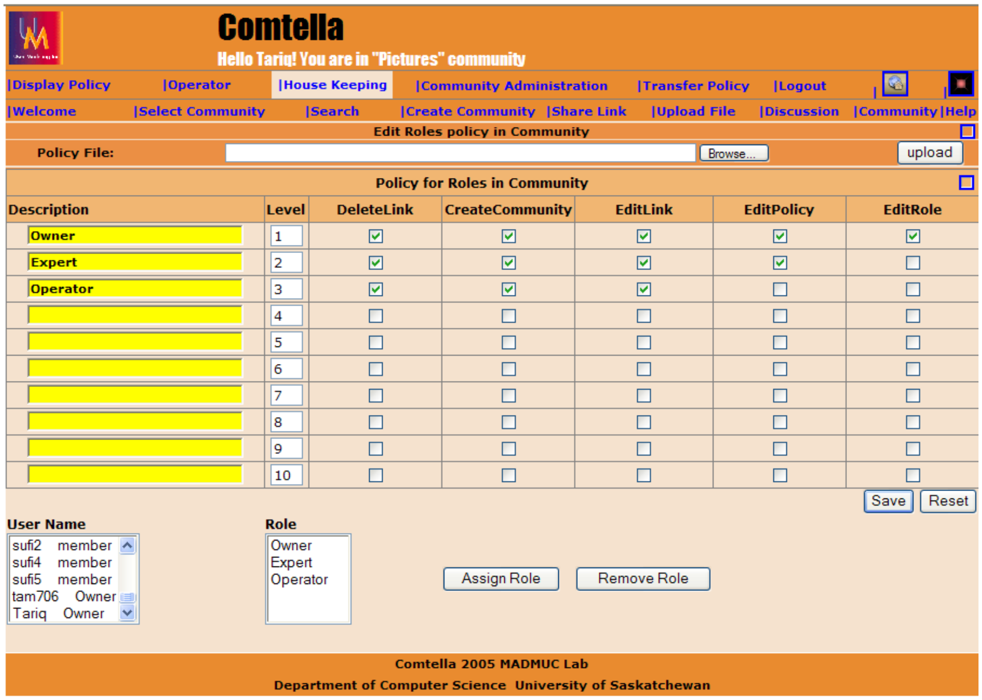
Like any organization, online communities can also manage their members in different roles. MCComtella uses role policies to allow community owners to share the burden of community management with deserving community members (see Table 4.5). The community owner should create policies to express the entitlements to these roles. Role policies define the conditions under which users with a given status can acquire a certain role and the accompanying rights and responsibilities. A community owner may designate a few members through either individual policy (by naming individuals) or through a selection-based policy (e.g. all gold-status members) to special roles, such as operators and experts. The moderator can assign special access rights to these roles such as editing and deleting resources (see Table 4.6). Role-based policies result in defining user groups based on their functional responsibilities such as expert, community moderator, and operator (see Fig 4.4).

Table 4.5 Role Based Policy Definition of Roles

<table>
<tr><th colspan="2">Role definition Policy</th><th>Description</th></tr>
<tr><td colspan="2">Community id: 1
Community Name: Picture
Node: http://kardam.usask.ca</td><td rowspan="7">Community owner can create new roles in community to share administration load with other members.</td></tr>
<tr><td>Level</td><td>Description</td></tr>
<tr><td>1</td><td>Owner</td></tr>
<tr><td>2</td><td>Expert</td></tr>
<tr><td>3</td><td>Operator</td></tr>
<tr><td>4</td><td>Member</td></tr>
<tr><td></td><td>...</td></tr>
</table>

Table 4.6 Role Based Policies Setting Access Permissions

Policy for Roles Permissions		√ Action allowed × Action not allowed				**Description**
Level	Description	Delete link	Create Community	Edit Policy	Edit Role	Based on roles user can set access rights for actions like delete link, create community, edit policy and edit roles.
1	Owner	√	√	√	√	
2	Expert	√	√	√	×	
3	Operator	√	√	×	×	
4	Member	×	√	×	×	
	....					



Edit Roles policy in Community

Policy File:

Policy for Roles in Community

Description	Level	DeleteLink	CreateCommunity	EditLink	EditPolicy	EditRole
Owner	1	☑	☑	☑	☑	☑
Expert	2	☑	☑	☑	☑	☐
Operator	3	☑	☑	☑	☐	☐
	4	☐	☐	☐	☐	☐
	5	☐	☐	☐	☐	☐
	6	☐	☐	☐	☐	☐
	7	☐	☐	☐	☐	☐
	8	☐	☐	☐	☐	☐
	9	☐	☐	☐	☐	☐
	10	☐	☐	☐	☐	☐

User Name	
sufi2	member
sufi4	member
sufi5	member
tam706	Owner
Tariq	Owner

Role

Owner
Expert
Operator

Comtella 2005 MADMUC Lab
Department of Computer Science University of Saskatchewan

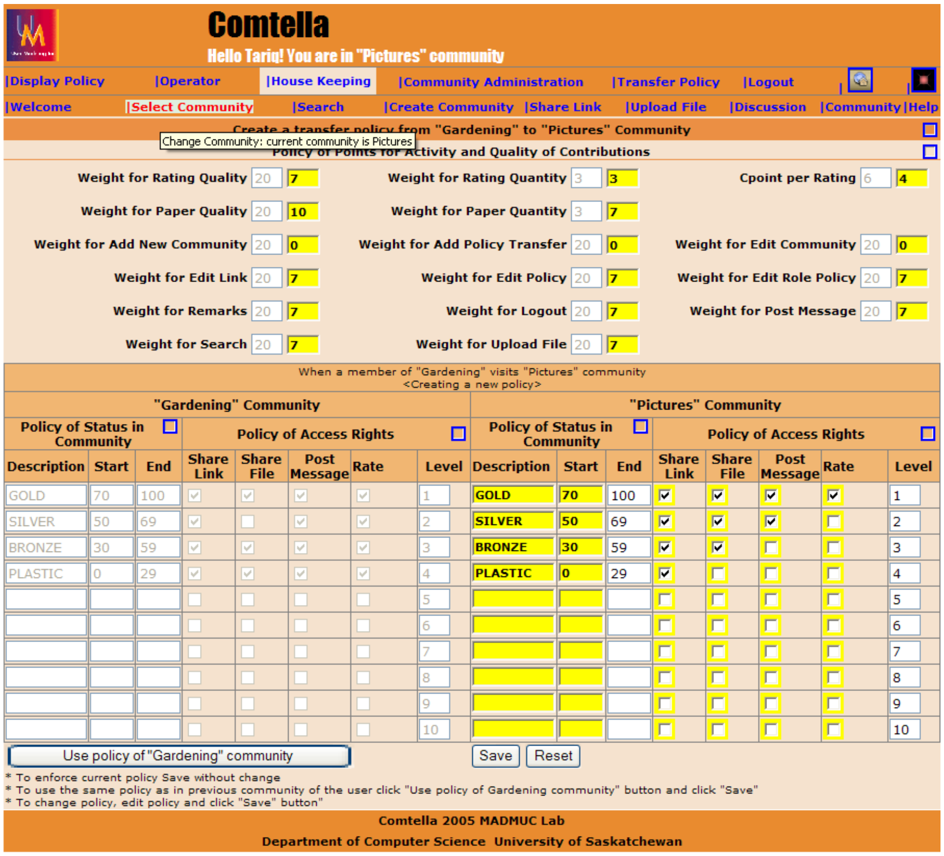
Figure 4.4: Interface for Editing a Role policy in MCComtella

4.5.3. Transfer Policies in MCComtella

The access, status and role types of polices presented in the previous sections define how to update the user model and what access rights to grant the user when she is working within the boundary of her community. When a user moves from one community to another, for example, by requesting access to a resource in a new community, a question arises about what status, role, rights and privileges, this user should have in the new community. To govern movement of users across two communities, the communities must have a contract/agreement about the status, role and access rights of visiting users. These contracts are called *transfer policies* and can be unilateral (e.g. the owner of the receiving community defines the policy according to which to treat visitors from other specific communities or in general) or bilateral (e.g. the two owners agree about mutual recognition of status, roles and rights) For example, the community owner may decide that visitors from the other community will be given automatically the same status in the new community or status with one level lower than the status they enjoy in their home community. In MCComtella these policies are unilateral (see Table 4.7). If a user wants to visit a new community (e.g. to read an article posted in this community), she has to send a request to owner of the destination community. The community owner sets a transfer policy after reading the policy under which user was working in her home community (see Figure 4.5). MCComtella allows three options for transfer policy to community owner: (I) enforce the current policy; (II) allow the policy of the previous community from where the user is coming; and (III) define a new policy for visiting users. The definition of a new policy can be achieved by using different approaches. One may be to show the community owner the policy of both communities and allow her to create a new transfer policy, as shown in Figure 4.5. In this approach community owners can define new statuses and their respective thresholds. Another approach may be to declare one of the status slots of the community equal to one or more slots of the other community from where a user is coming.

Table 4.7 Transfer Policy from Community "Gardening" to Community "Indoor Plants"

Status Permissions policy √ Action allowed × Action not allowed					Description	
Level	Description	Share link	Share File	Post	Rate	Transfer policy is used when a use visits a community to create a user model. It can be same as the status policy of current community or owner can set a new policy.
1	Gold	√	√	√	√	
2	Silver	√	√	√	√	
3	Bronze	√	√	√	√	
4	Start	√	×	√	√	
	....					



Create a transfer policy from "Gardening" to "Pictures" Community □

Policy of Points for Activity and Quality of Contributions □

Weight for Rating Quality	20	7	Weight for Rating Quantity	3	3	Cpoint per Rating	6	4
Weight for Paper Quality	20	10	Weight for Paper Quantity	3	7			
Weight for Add New Community	20	0	Weight for Add Policy Transfer	20	0	Weight for Edit Community	20	0
Weight for Edit Link	20	7	Weight for Edit Policy	20	7	Weight for Edit Role Policy	20	7
Weight for Remarks	20	7	Weight for Logout	20	7	Weight for Post Message	20	7
Weight for Search	20	7	Weight for Upload File	20	7			

When a member of "Gardening" visits "Pictures" community
<Creating a new policy>

"Gardening" Community									"Pictures" Community								
Policy of Status in Community □			Policy of Access Rights □						Policy of Status in Community □			Policy of Access Rights □					
Description	Start	End	Share Link	Share File	Post Message	Rate	Level		Description	Start	End	Share Link	Share File	Post Message	Rate	Level
GOLD	70	100	☑	☑	☑	☑	1		GOLD	70	100	☑	☑	☑	☑	1
SILVER	50	69	☑	□	☑	☑	2		SILVER	50	69	☑	☑	☑	□	2
BRONZE	30	59	☑	☑	☑	☑	3		BRONZE	30	59	☑	☑	□	□	3
PLASTIC	0	29	☑	☑	☑	☑	4		PLASTIC	0	29	☑	□	□	□	4
			□	□	□	□	5					□	□	□	□	5
			□	□	□	□	6					□	□	□	□	6
			□	□	□	□	7					□	□	□	□	7
			□	□	□	□	8					□	□	□	□	8
			□	□	□	□	9					□	□	□	□	9
			□	□	□	□	10					□	□	□	□	10

* To enforce current policy Save without change
* To use the same policy as in previous community of the user click "Use policy of Gardening community" button and click "Save"
* To change policy, edit policy and click "Save" button"

Comtella 2005 MADMUC Lab
Department of Computer Science University of Saskatchewan

Figure 4.5: Interface for editing a transfer policy in MCComtella

4.6 Architecture of the Policy Framework (PF)

In MCComtella each community uses a policy execution framework, which interacts with the user model instances for each community. It consists of the following components: (see Figure 4.6):

- A "shared view" used for all context and user data, both raw data as well as calculated user attributes;
- A set of user policies governing the community, each specifying the input data (about the user and context), a process and output data;

- An execution mechanism running in a loop which selects an appropriate policy for the current user request and context and executes its process.

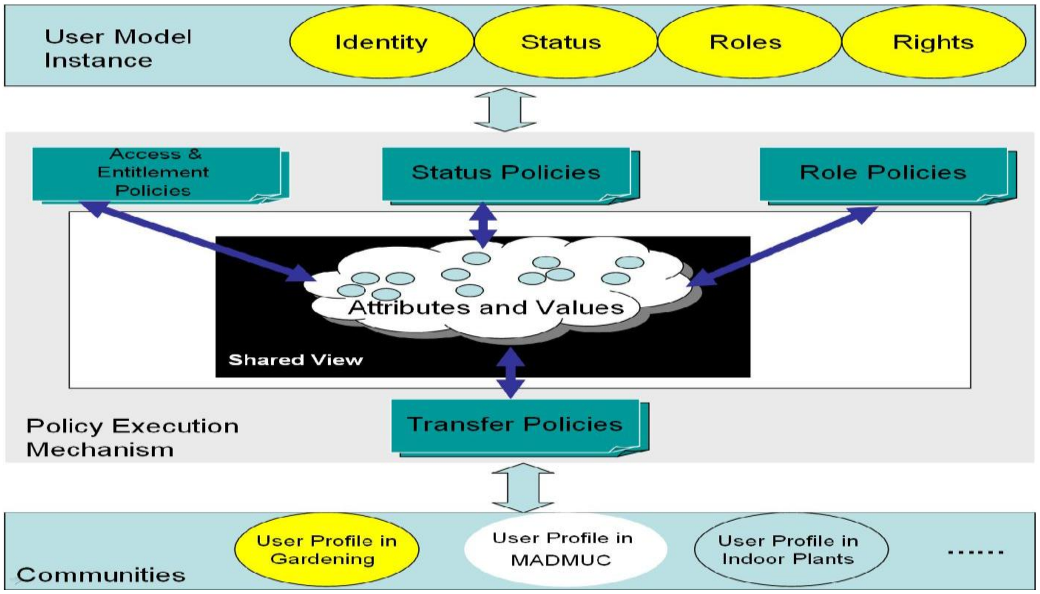


Figure 4.6. Policy Framework (PF)

The policy framework is responsible for the calculation, discovery and security on reading and writing of user attributes on the '*shared view*'. The Policy Execution Mechanism fetches a user profile fragment (a record of user activities, as well as user role and status) from a particular community, and calculates the new role and status of the user in the community according to the role and status policies of the community thus creating a user model just in time. It also determines the user access rights and privileges, as well as interface adaptations. The policy framework ensures that the relevant policy is selected depending on the user request (which arrives on the shared view). The invoked policy in turn picks the required user data items (either raw data or user model data computed as output by other policies, in the same community or requested from other communities). There are many different policies in the set, which can be seen as managing different levels of decisions. For example, there are high-level policies that compute the role and status of the user in the community, using user data received from other communities (which is either raw participation data or data computed by other policies). Lower-level policies control the user access rights using data about the user role or status computed by the higher-level policies. In this way, the framework provides both personalization and a simple security layer to protect against unauthorized users and actions.

A policy has three components: input, process and output. The input is either raw user data or data computed as output by other policies. For example, the input of a policy controlling user access to a community can be a user action attempting to access an item shared in the community. As another example, the input of a user access policy controlling user access to a community can be an action of a user attempting to rate a posting in the community. We call such raw data indicating user intentions a "user request". A request consists of three parts: the subject, action and resource (Merrells, 2004) (OASIS, 2005) (Seth, 2004). Here "subject" is a primary identity key hosted at either a shared identity provider (Erik & Vullings 2005) or by one of the federation of identity providers. This identity is placed on the shared view where other policies can use it as input, for example, to fetch the user attributes hosted in the user database from both the current community (that is receiving request) and from any other community which has data about this user. For example, an access policy will use the user id as input and will retrieve the user attributes relevant to granting the user access rights to particular resources. The process of this policy will check the user attributes, in particular the user's role, and adapt the interface to allow the functionality for which the user should be granted access. A status participation-measure policy will use the user id and the user request as input and will retrieve the user attributes relevant to the user participation from the user model. The process of this policy will recalculate the user participation metric including the user intention action and will generate a new user participation value in the user model as output. These user attributes can be inputs for another status policy, for example, one that decides what status to grant the user in the community based on her participation measure.

The process of a policy involves the algorithm that computes in context the output user model data or makes an adaptation decision. The context is defined by the user data present at the moment on the "Shared View". The process is executed by policy execution mechanism which retrieves the user profile, the data required by the policy as input, and places it in the Shared View. The policy framework execution mechanism then computes the policy output data using the available input and current context data from the shared view and makes a decision, for example to allow / disallow the request or to adapt the functionality or interface. For example, the process of an access control policy distinguishes between new users and local users (whose profiles are stored at the community). For a local user it retrieves the location of her user model, which becomes the output of the purpose and either grants or denies access depending on the role of the user. For new visitors it calls the appropriate transfer policy whose inputs match the user request and the current context and produces its output. For example, the process of the transfer policy (using the user id as input) requests information from all other collaborating communities that have stored a model of this user and generates a local user model for the new user, which contains her status and role according to the mapping algorithm assigned by the community owner. This data will then be used as input by the community's rights and status policies that decide about the user's rights and privileges.

4.7 Summary

In this chapter describes the proposed approach for using policies for user modeling and adaptation in online communities. It provides an account of policies within a community and across communities. It lists the different policies implemented in MCComtella such as status policies, role policies and transfer policies. The chapter discussed at conceptual level, how these policies are used to update the user model and presented the policy framework implemented in MCComtella.

The new approach with explicit policies implements open user modeling approach which allows end users to define how user modeling and adaptation is going to be done. The policies provide explicit rules according to which the user model is updated and adaptations are made. There are several advantages in using explicit policies to define the user modeling process. The advantages of using open policies for user modeling and adaptation in online communities are:

- Empowering the users to manage the adaptation in the community. Even though only community owners can define and modify the policies in their communities, any user can create a new community and be a community owner.

- Explicitly assigned roles for users lead to a more sophisticated user model, representing the context, purpose, trust and reputation of users within and across communities.

- People who are members of other communities will feel more comfortable about the access and other rights of ‘strangers’ since they are allowed to visit their community only after transfer policy negotiation between the community owners.

- Transfer policies allow for interactions between communities result in exchange of both users and contents, which is otherwise not possible.

- One problem in many learning communities is the ‘cold start’ (Denaux et al. 2004) (Sun &Vassileva 2006) where the system fails to provide adaptation due to the lack of information about users when they first visit a community. With transfer policies the community does not have to wait for the accumulation of user information to offer customization and adaptation. Transfer policies allow acquiring user model data about the previous experience of the user from other communities.

- The main advantage of a decentralized user modeling approach is taking into account the local context in the integration of possibly inconsistent user data fragments. This is very hard to do automatically, since it requires understanding the context in which these data fragments were created. Our policy-driven approach solves this problem by putting the human in the loop in creating transfer policies. Thus the user enforces the local context on any adaptation decision. It can also be viewed by users who can understand how user modeling and adaptation happens in the community. In this way a human-centered way of user modeling in context takes place.

Chapter 5

Evaluation

The proposed policy-based approach for user modeling in online communities puts a "human in the loop" to create and edit the policies that govern what user modelling and adaptation processes will take place in the community. The "human in the loop" can be any user -- not a programmer, or a specially trained operator. In order for this approach to succeed it is crucial that users can accept and comprehend the notion of policies and learn how to set new community reward policies, edit existing policies, and how to create transfer policies for users visiting from other communities. While a general survey could collect some initial data about the user's attitudes on these questions, it will not be grounded in experience with a real system. The MCComtella is used in the study as a system demonstrating the policy based user modeling approach "in action". As with any software system for end-users, the design of the Graphical User Interface (GUI) for viewing and editing policies is important to ensure user acceptance, understanding and ability to work with policies. Therefore, the evaluation has to take into account also the usability of the GUI, even though designing the GUI is not the main goal or contribution of the proposed approach.

Therefore the evaluation focuses on the user comprehension of the notion of policies, their importance in the context of multiple communities, and on the usability of the GUI provided for viewing and editing policies.

5.1 Methodology

To meet the goals of the evaluation (according to the hypothesis), the following methods were used:

- ***GUI usability (in terms of user ability to perform certain tasks with it):*** To evaluate the usability of the GUI for creating and editing the user status/transfer/role policies in MCComtella, the study evaluates the user's ability to read and edit policies using the Graphic User Interface (GUI) in performing tasks according to several scenarios.

- ***Acceptability of policies:*** To evaluate the user acceptance of the concept of polices to manage user model and adaptation in online communities and moving user data across communities, the study includes observations of users playing the role of community owners completing scenarios with different tasks. This was followed by de-briefing sessions where the users had to explain the policies they had created and reflect on the process and how it met their intentions.

- ***Overall satisfaction with MCComtella as an online community framework:*** To evaluate the overall users' satisfaction with the functionality and usability of the MCComtella framework system, we used a questionnaire.

5.2 Participants

The participants were recruited through email invitations to different computer labs at the University of Saskatchewan Campus, and to the gardening community in the University Residences. The goal was to have participants with various backgrounds since any user may wish to create his/her own community. There were twelve participants in the study; each was paid $10 for taking part in one-hour long experiment. The age of the participants (7 males and 5 females) ranged from 23 to 29 years. Six participants were graduate students in the Computer Science Department of the University of Saskatchewan. Four participants were graduate students from the School of Engineering, Veterinary Sciences, Soil Sciences, and Chemistry. Two participants were undergraduate students at College of Arts and Science, and College of Commerce. Participants were invited to do the evaluation in the MADMUC lab.

We used a questionnaire to find out the users' backgrounds, experience with online communities and their willingness to create their own community, define user policies and their overall evaluation of the Comtella framework. Each of the participants has accounts in more than one online community. Nine out of twelve participants have more than four accounts and seven out of twelve participants have five or more accounts

5.3 Procedure

We used a questionnaire to find out more about the users' backgrounds and their reaction to the idea of creating their own community, defining user policies and their overall evaluation of the MCComtella framework. A 15-minute introduction was given of the notion of policies, the MCComtella communities and the interface screens related to editing policies. Then we asked the participants to go through five scenarios related to creating a community, defining user policies of different kinds and modifying the policies. We observed and recorded each user's performance, including the degree of successful completion of the given task. Then we tested their understanding by asking them to write down in their own words the community polices created by them. After each task in the scenario, we asked both general and specific questions regarding the task, for example, regarding the usability of the MCComtella web pages, details of the form/web page widgets, the GUI ease of use, the user's comprehension of policy from the screen contents and their overall experience. This study was approved by the behavioural research ethics committee at the University of Saskatchewan (BSC03-1046).

We used objective measures (the error rate) and subjective performance measures, (observation of the performance of users, the understanding of the community policies, the user satisfaction and comprehension of the GUI for policy editing). This scenario-based study provided information about the readability of policies, usability of the policy-editing interface, the users' conception of open user policies, their preferences for tools to author community policies and the learning curve of system.

5.4 Scenarios

We defined five scenarios with tasks that required creating and editing policies. The overall objective of these scenarios was to verify the general acceptability of policies to control status, user transfer and community roles management. We asked the participants about the usability; ease of learning and capabilities of Comtella as a community system.

To evaluate the acceptability of user policies we asked two types of questions. The first type is related to the overall importance of policies in an online community. The second type of questions tests the acceptability, comprehension and readability of the policies in the current implementation of the Comtella framework. We had two purposes for asking these questions; first to establish if the users perceive a need for policy-based user modeling, and second, to verify the usability of the current implementation of the policy-editing GUI and whether it is able to correctly communicate the idea of polices in online communities to participants.

The full scenarios and the user questionnaires that were completed after each scenario are presented in Appendix B. An observer was recording the user performance and was available to answer the users' questions regarding the tasks to be performed (see Appendix A to find the observer evaluation form).

In the first scenario the participants played the role of community owner and performed one task: to create a new community. The participant automatically becomes the owner of the community, which allows the participant access to actions to set the status- and role- policies for of the community and the transfer-policies for users coming from other communities.

The goal of the second and the third scenarios was to verify the participants' understanding of user status, how a policy expressed the rules for updating the user status, and the usability of the GUI screen that allowed creating and changing a status policy. In both of these scenarios the participant played the role of community owner. In the second scenario, the participant had to create a status policy for the new community including a status update policy and a status level policy. The third scenario asked the participants to change an already existing set of status policies by modifying the previous values of weights, thresholds and rights, rather than creating new policies. To accomplish this, the participants had to use the same GUI screen as in the previous scenario.

The fourth scenario aimed to verify the participants' comprehension of the notion of transfer policy and specifically the transfer of user status. This scenario is fairly complex due the amount of information and actions that can be performed in one web page. In this scenario the participants were asked to set a transfer of status policy from "Gardening" to the "Pictures" community. The participants were in the role of the community owner of the "Pictures" community and performed three tasks, corresponding to different possible transfer policies that would be applied to visitors coming to the "Pictures" community. The first task was to allow visiting members coming from the "Gardening community" (i.e. visitors who have set the Gardening community as their default community), to use the status policy of their original "Gardening" community as a transfer policy to the "Picture" community (that means that they would enjoy the status and privileges they had in "Gardening" during their visit in the "Picture" community). The second task was to allow visiting members to use the status policy of "Picture" community as a transfer policy (that means that the visitors will be treated according to the status policy of the "Picture" community and they may not have any status and privilege during their visit, if they haven't visited before). The third task was to create a new transfer policy for users coming to the "Pictures" community from the "Gardening" community (applying a “conversion” agreement for their status and privileges, supposedly negotiated with the owner of the “Gardening” community).

In the fifth scenario the participants performed just one task - to create the role policy of the new community. The participants were again in the role of an owner of the community, who creates different roles in the community, assigns rights associated with each role and assigns those roles to particular community members. For example, an “expert” member can be allowed to delete a link, create a new community, edit a link and edit a role. The objective of this scenario is to verify the participant’s comprehension of the roles in community and policy-based roles management. We also wanted to verify the usability of the GUI for role-policy editing.

After the experiment, a questionnaire was administered to the participants. Two types of questions were asked, to evaluate the need for policy and the acceptability of user policies. The first type of questions asked users about their opinions of the overall importance of policies in an online community (see Appendix B). The second type of questions tested the comprehension and readability of the policies in the current implementation of the MCComtella framework (see Appendix B). Also questions about the overall reaction of the user to the MCComtella system were asked (See Appendix C).

5.5 Results

This section presents the results of the evaluation study. It presents performance of participants in various tasks that reflects usability of Comtella GUI. We also collected the feedback of participants regarding acceptability of policies and GUI. A summary of user’s responses is given in appendix D.

5.5.1 Task performance in the five scenarios

The participant's performance was observed and classified in two categories: "performed without help" and "performed with help". The results are presented in Table 5.1. Further on, the "performed with help" cases were classified into two sub-categories: "Performed with explanation" where the participants received a verbal explanation or reminder about the next step and "Couldn't perform" where the participant was unable to complete a step in task and it had to be demonstrated to the participant. The performance results encoded in this way are presented in Figure 5.1.

Table 5.1 Task performance in the five scenarios

Create Community		Create Status Policy		Edit Status Policy		Create Transfer Policy								Create Role Policy		
Task1		Task2		Task 3		Task4.1			Task4.2			Task4.3		Task5		
Help	No Help	Help	No Help	Help	No Help	Help	Couldn't perform	No Help	Help	Couldn't perform	No Help	Help	No Help	Help	Couldn't perform	No Help
0	12	4	8	5	7	5	1	6	2	1	9	1	11	1	1	10

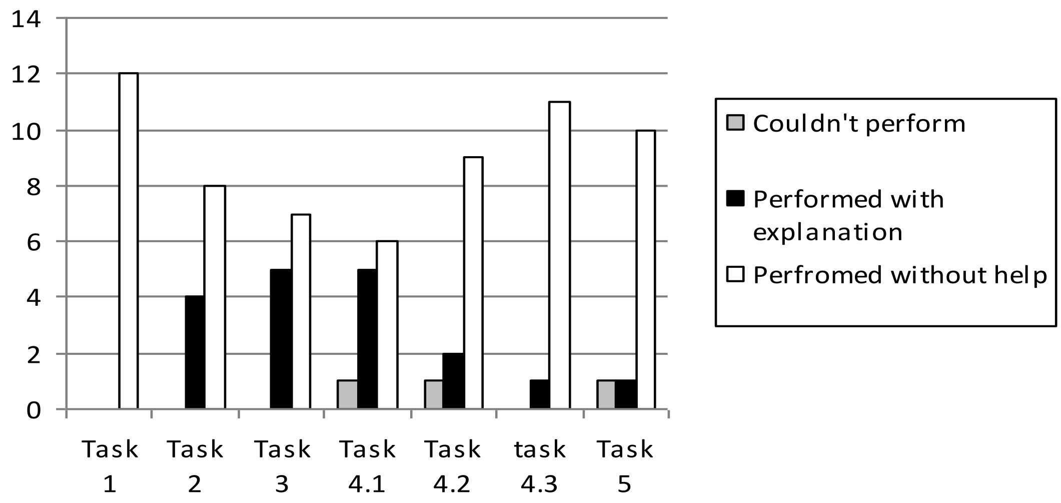


Figure 5.1. User Performance in Tasks

Tasks performed with explanation, couldn’t perform, and without any help

The results show that most of the users performed most of the tasks successfully (Table 5.1). The tasks related to scenarios 2 and 3 (creating a status policy - Task 2 and editing a status policy - Task 3) caused difficulties to about one-third of the participants. The number of requests for help to perform task decreased as user progressed through the 4th scenario (from task 4.1 to 4.3). Fig. 5.1 shows that the observer intervened only three times to partially perform the task for a user. Not surprisingly, the fourth scenario (creating and editing a transfer policy) caused most difficulties and one user was unable to perform the tasks 4.1 and 4.2. Interestingly, the participants performed better on task 4.3 than on task 4.1 and 4.2. The task 4.3 requires the user to create a new transfer policy, so it could be considered as more complicated than the previous two tasks, which required the user to just check a box (stating that either the status policy of the origin community or the status policy of the current community will be used as it is). Perhaps the users had already gained experience in creating a new status policy and did not find it difficult to create a new transfer policy, different from the policies of the two affected communities. Interestingly, another user was also unable to perform the last task in Scenario 5 creating a role policy, which was similar to task 2 (creating a status policy) and actually simpler than tasks requiring the creation of a new policy, task 2 and task 4.3.

After performing all the tasks, the participants were asked to give their overall evaluation of the interface (screens), ease of learning and capabilities of the Comtella system using a Likert scale from 1 to 9, where 1 means bad/ difficult and 9 means good/easy (see Appendix C). The user's responses are summarized in Tables.5.2, 5.3 and 5.4.

Table 5.2 Creating Status Policy Screen (User's feedback) n=12

	Very poor -2	Poor -1	Satisfactory 0	Good 1	Very Good 2
Overall				75%	25%
Usability			8.33%	25%	66.66%
Reliability (crashes etc.)			8.33%	33.33%	58.33%
Policy File Upload				33.33%	66.66%
Adding and removing Action				33.33%	66.66%
Overall readability of policy				41.66%	58.33%
Readability of XML file				58.33%	41.66%

Table 5.3 Creating Transfer Policy Screen (User's feedback) n=12

	Very poor -2	Poor -1	Satisfactory 0	Good 1	Very Good 2
Overall				50%	50%
Usability			8.3%	33.3%	58.3%
Reliability (crashes etc.)				33.3	66.6%
Overall readability of policy				50%	50%

Table 5.4 Creating Role Policy Screen (User's feedback) n=12

	Very poor -2	Poor -1	Satisfactory 0	Good 1	Very Good 2
Overall				16.7%	83.3%
Usability				16.6%	83.3%
Reliability (crashes etc.)				25%	75%
Policy File Upload				8.3%	66.6%
Adding and removing role				16.6%	83.33%
Overall readability of policy				8.3%	91.6%
Readability of XML file				25%	75%

The users' feedback of the graphic user interface (screen layout) is positive overall (see Fig. 5.2). The Comtella interface organizes policy editing tasks that would occur naturally together (as expressed in the scenarios and tasks) into one screen. Within one screen the form fields were divided into logical groups and contrasting colors and layout. If we compare the usability feedback for the status policy, transfer policy and role policy for 'very good' it is 66.66 %, 58.33% and 83.3% respectively (see Table 5.2, 5.3 and 5.4). This means that the screen for the transfer policy requires improvement, so that the community owner can easily set a transfer policy from one community to another. Similarly the "very good" feedback for the overall readability of status, transfer and role policies is 58.33%, 50% and 91.6% respectively (see Table 5.2, 5.3. and 5.4). This may result from the complex nature of transfer policies that makes it hard to create a simple transfer policy editing screen. This problem could be solved by providing a separate screen for each transfer policy option: use policy of old community (from where user is coming), use policy of current community (where user wants to join) or create a new policy for visiting user. The current interface design offers all of these options in one screen.

Scales used for the questions:

- **Screen layouts were helpful:**
  1 – never; 9 – always
- **Sequence of screens:**
  1 – very confusing; 9 - clear.

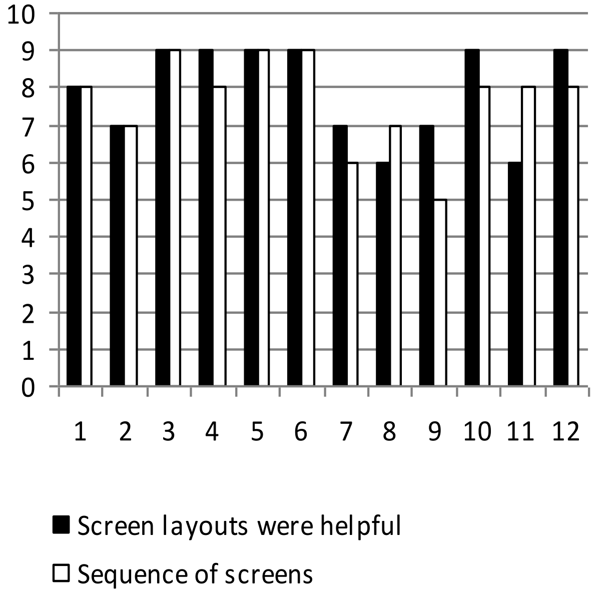
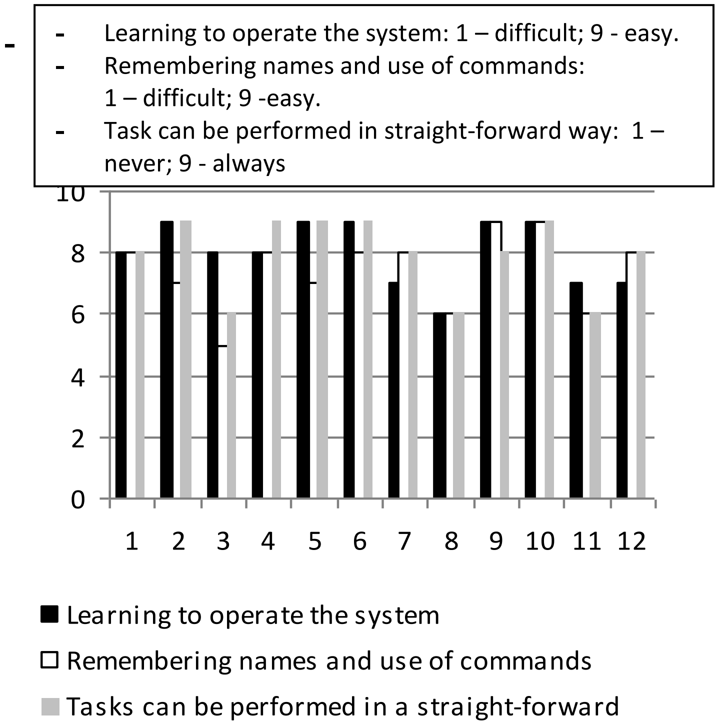


Figure. 5.2. User feedback about the Comtella interface (screens).

Learning the system and performing tasks was judged as easy by most of the users (see Fig. 5.3). Most of the participants said that tasks can be performed in a straight forward way. However, it wasn’t so easy for some of them to remember the names and use of commands; yet the lowest rating on this question is 5 (out of 9), given by only one user. This problem can be rectified by providing in context help by a popup screen and making screens across the system more uniform.



-
- Learning to operate the system: 1 – difficult; 9 - easy.
- Remembering names and use of commands:
  1 – difficult; 9 -easy.
- Task can be performed in straight-forward way: 1 – never; 9 - always

Figure 5.3. User feedback about MCComtella learnability.

The participants also found the MCComtella system reliable, with good response time (see Fig. 5.4). Most of the participants were satisfied in with the speed of the system. Comtella is designed on sound application architecture principles using model view design pattern.

- **System speed:** 1 - very slow; 9 – very fast.
- **The system is reliable:** 1 - never; 9 - always.

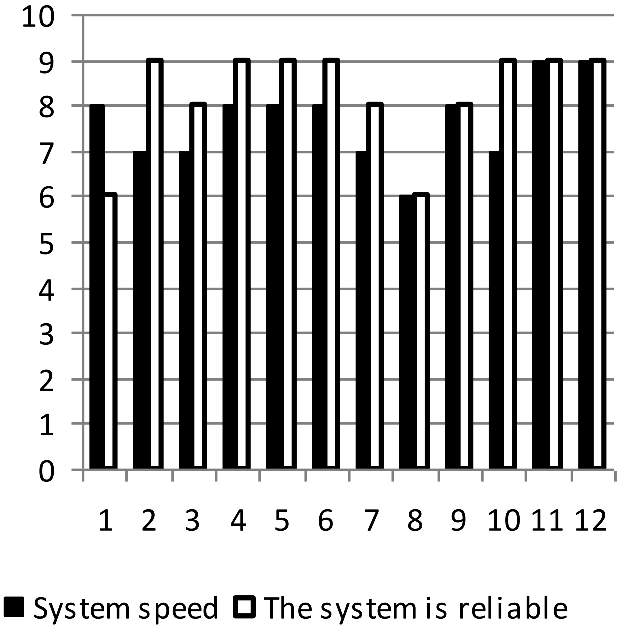


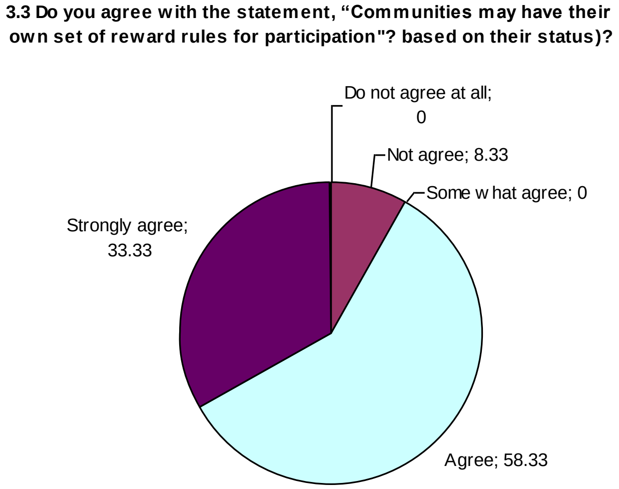
Figure 5.4. User feedback about Comtella capabilities.

Overall, from the evaluation results it seems that the current implementation of the GUI for policy editing is usable and learnable, even though there is room of improvement in the interface design for complex policies, e.g. the transfer policies. A possible approach is a "wizard" that guides the user in developing logically related and consistent policies.

In this section the results related to the user's performance and usability of Comtella interface (the screens related to creating and editing status, transfer, role and reward policies) were presented. The next section reports the results from the questionnaire that tested the user acceptability of the policy-based user modeling approach.

5.5.2 Acceptability of Policies

The questionnaire asked the participants if they supported the idea that communities may set their own policies for reward (See appendix B for Questions). One third (33.33 %) of the participants agreed strongly and 58.33% agreed with the idea. The remaining 8.33% were neutral or moderately negative (see Fig 5.5). However, 91.67% supported the idea that the community owner may set the reward mechanism for the community. All participants supported the idea of limiting user access based in their status and role in a community (see Table 5.5). This feedback from the participants supports our assumption that the status policy should be maintained at community level.



3.3 Do you agree with the statement, "Communities may have their own set of reward rules for participation"? based on their status)?

Figure 5.5 The need of local status policies for communities.

Table 5.5. The concept of user status policy and transfer policies

	Do not agree at all	Not agree	Somewhat agree	Agree	Strongly agree
The community owner should be able to set reward rules for his/her community members.	0	0	0	1	11
Would you support the idea of limiting users' actions in the community based on their status (for example limiting who can read, share and comment based on their status)?	0	0	0	9	3
Do you agree with the statement, "Communities may have their own set of reward rules for participation"? (Based on their status)?	0	1	0	7	4
If you find a policy of some other community suitable for your community, would you request the XML based policy document from that community owner?	0	1	0	10	1
When a user moves from one community to another, do you think his/her status should be transferred to the second community?	0	2	3	7	0
Do you think the community owner should be able to set rules for transfer of user from one community to another?	0	0	0	7	5
If you found a Role Policy of some other community appropriate for your community, would you like to request the XML based role policy document from that community owner?	0	1	1	8	2

When asked a question about the transfer of status from one community to another community 58.33% participants replied that they agree with this idea and 25 % replied they somewhat agree with this idea (see Fig. 5.6. Probably the reason is that user consider the transfer of status as appropriate across communities that share something (common interest, goal, ideology or demographics), but not appropriate in general, for two random communities. All participants supported the idea that the transfer policy should be set by the community owner (Table 5.5). Some of the remarks from participants supporting a portable user model were:

"So I don't start from scratch, provided that both communities (current and new) are at the same level of proficiency/rating system". and *"Save your time to start over again*"

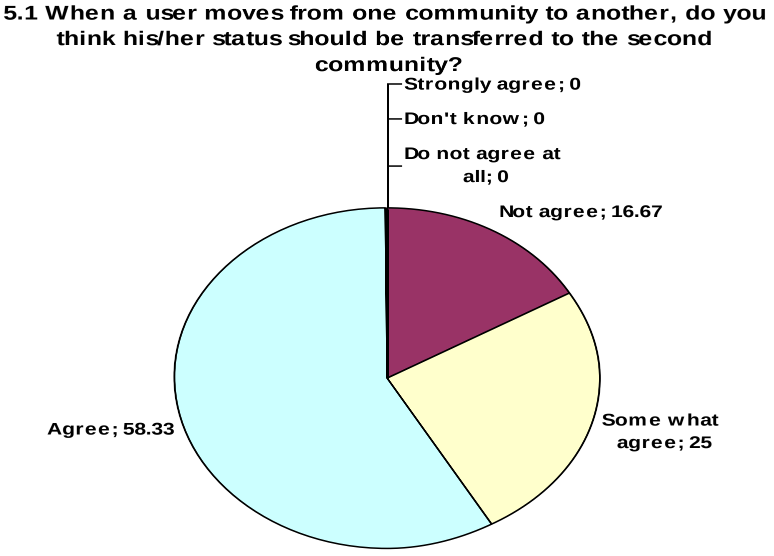


Figure. 5.6. Transfer of user status from one community to another.

When the participants were asked if they would request a community owner of another community to send them a policy file they found was suitable for their community, 91.7% of the participants responded that they would do that (see table 5.5). This feedback supports our assumption that policies should be portable across communities and they should be in human readable and editable format. The responses indicate that the users would like to reuse the policy from another community.

Participants in this study are of computing and non computing backgrounds. Some had little but most had no experience using the MCComtella online communities. Nevertheless, all participants learned the system quickly. The participants' comprehension that policies are a tool for managing communities also evolved rapidly. They were able to edit and explain the default policies from Comtella screens in their own words (see Table 5.6). They also expressed their opinion about who should be responsible for setting rules for policies (see Table 5.5). They expressed their views about to control the user status and roles and showed their interest in reusing the policies of other communities instead of writing their own (See table 5.7). They also supported the idea that transfer of policies will help users, by not starting from scratch when they join new communities. These factors show that participants appreciated the possibility of using policies to manage online communities. The questions that tested the participants' comprehension of policies are presented in Appendix B. The resulting user answers to these questions are presented in Appendix D.

Table 5.6 Scenario: Edit a "Role Policy" of the community (User's feedback)
Q 6.4 The Comtella "Role Policy Screen" offers a default Role Policy. Please, describe in your words what rights the Role Policy assigns.
Owner can delete link, create com, edit link, edit policy, edit rule. Expert can as above. Operator can as above….
The rights offers are as follow delete link, create new community, edit an existing link, editing an existing policy and also edit the role of an existing member
The ability to carry out roles deleting, creating a community, editing a link, editing a policy and editing a role
ability to delete offensive material, create new community and branch off, change their own status
The rights allow the member to manage the community based on their roles.
Delete or edit link, create community, edit policy and role
Delete link, create community, edit link, edit role
The owner expert and operator have different level of roles, the owner has all the roles and he can assign his friends to have these roles. The role of operator and expert can be changes otherwise.
The right to delete a link posted by other users, to cr4eate a community, to edit a link posted by other users, to edit an existing policy and to edit existing roles
assign different level of users to do different work
based on his/her status/role a user can perform various tasks as editing policy or roles, links etc delete links and create community.
delete link, create community, edit link, edit policy

Table 5.7 Suggestions Edit Status Policy

Q-4.3 Do you have any comments or suggestions regarding editing existing policies?
I liked the idea of requesting a xml based policy doc. From other community owner will enhance the diversification the policy rules.
Editing strategy is easy for anybody that knows the basics of computer
No need for external program to edit and upload policies. Complicates step, may confuse user
Changing the weight of rewards for the action is complex. It would be easy to directly change the content in the selected action tags and weight box
Good luck

5.6 Summary

This chapter provides the results of the evaluation study of the proposed open policies based approach for online communities. The hypothesis was that the readable and editable user policies for communities are easy to comprehend and are acceptable for users. To test this hypothesis we used the MCComtella multi-community framework and its GUI that supports the user in creating and editing policies for managing, transferring and adapting user models between the communities existing within the MCComtella framework. The results of the study showed that the users were generally able to comprehend the notion of policies and were able to create new policies and edit existing ones, using the MCComtella GUI designed for this purpose. A limitation of the evaluation is that the results obtained are based on the tools that the users had access to (the GUI and MCComtella), but they can give an indication of the viability of user-editable policy approach for user modeling and adaptation in general (not restricted to MCComtella) online communities.

Chapter 6

Contributions and Future Work

This thesis proposed a policy based-approach for user modeling that allows explicit declarative representation of the user modeling and adaptation process in terms of policies, which can be viewed and edited by users. This policy-based user model framework is implemented in the MCComtella community framework, which allows hosting multiple communities, creating new communities by users, and movement of users across communities. The MCComtella community framework was developed as part of this thesis work. Next, an explanation of how the proposed policy-based user modeling approach provides a solution to each of the original problems is given.

6.1 Summary of the problems solved in the thesis

The thesis addresses three main problems in the area of user modeling and adaptation in the context of online communities stated in the introduction chapter:

4) Dealing with *unique and changing user modeling needs* of online communities.

5) *Involving users in the user modelling process.*

6) *Interoperability* of user models across different communities.

A summary of these problems and how the proposed policy-based approach solves them is provided below.

1) Dealing with *unique and changing user modeling needs* of online communities

Providing rewards for participation and allowing users to grow their status is a very important feature present (either explicitly or implicitly) in all successful communities. Currently, the rewards are typically static and built in the code of the system by the designer (usually the community owner, e.g. in Slashdot). However, the experience of Wikipedia administrators (Forte & Bruckman, 2008) shows that there is a need to adjust the rules for giving certain users higher status and particular responsibilities (roles) flexibly following a democratic process to meet the changing needs of the community. For this purpose, the process of user modeling to measure user participation and the corresponding adaptation in the community (to provide rewards) has to be open and manageable by the users. These can be users that are granted special responsibilities, e.g. owners or community-elected moderators, but nevertheless they are users, not system designers.

However, the current state of development of user modeling techniques does not allow for such flexibility. The user models (their structure, representation, and methods for updating) and the ways they are used for adaptation are currently defined at design time and cannot be changed at run time. Furthermore, they are usually part of the system code and not comprehensible for users. There is research in “open learner modeling” (reviewed in Chapter 2) that aims at revealing the content of the user model to the learner to stimulate reflection and better learning, but so far there hasn’t been any work on opening the methods for updating the model or the methods for adaptation to the user.

The proposed approach makes the updating of the user model and the adaptation process open to the user (by expressing these processes in policies) and allows users not only to view these processes, but also to change them, by editing the policies. Thus the community owner can react to changing needs of his/her community and adapt the reward policies of the community.

2) *Involving users in the user modelling process.*

By empowering users to create and edit policies, the second problem stated in the introduction is addressed, involving users in the user modelling process. The evaluation of the implementation in the MCComtella community (presented in Chapter 5) of the proposed policy-based user modeling approach shows that the notion of policies is understandable and acceptable for users, and the provided GUI interface allows them to successfully complete several typical scenarios that they would have to do as community owners.

3) *Interoperability* of user models across different communities.

Current online communities are “walled gardens”, with no exchange of users and user model data across communities. This leads to a major inconvenience for users – they need to remember different passwords and to build up their presence in different communities from scratch. While some social network application developers have recognized this problem very recently and have developed some APIs (e.g. Facebook) and standards for sharing data (OpenID, OpenSocial), this data is mostly trivial (e.g. user name, the names of his/her friends, as well as their updates). No compiled participation data or data about user status, role and privileges is shared. The platform developed for this research (the MCComtella framework) demonstrates what kinds of adaptation are possible when online communities are able to share user model data. Surely this sharing needs to be controlled to ensure privacy for the community members, therefore the need for policies in user data sharing emerges naturally.

The proposed solution to facilitate interoperability of user modeling across different communities is to create transfer policies – explicit policies that map user status or participation data accumulated in one community to status or participation in another community. This approach provides a solution to another outstanding problem in the area of user modeling – how to implement practically a scalable solution for purpose-based, decentralized user modeling that combines user model fragments and interprets them in context to make a decision for adaptation just in time? While the purpose-based decentralized user modeling approach has been proposed back in 2003 (Niu et al., 2003) , there has been no working solution proposed on how to create sufficiently large libraries of purposes that capture different possible contexts (what kind of user data is shared), representation of user data (e.g. participation can be represented with *points* in one community and with *active time* in another) and different possible sources (e.g. from which community the user data is coming?). This thesis provides an answer to this question – to involve the users in creating the purposes. By placing a "human in the loop", it becomes possible to create policies (which implement purposes for user modeling and adaptation), that are suitable for the particular context (stage of development of the community, dynamics etc.) and the particular goal of the user.

6.2 Main Contributions

This thesis makes several contributions to the areas of user modeling and the design of online communities.

1) User Modelling

- a novel open user modeling approach where users can view and change the process of updating the user model and the process of adaptation,
- a practical approach for decentralized user modeling; the human editable policies implement purposes for user modelling,
- one of very few existing approaches for user modeling and adaptation for online communities.

2) Design of online communities – the MCComtella and MNMCComtella platforms

- supporting movement of users across multiple communities,
- supporting explicit reward policies that are open and editable by users,
- useful for practice to support communities and future research.

6.3 Future Directions

The Comtella community framework with policy-based user models provides a platform to study the dynamics of online communities. We can envisage several useful studies that can be done in the future.

Study of reward mechanisms in a single community.

Comtella has been used for the study of reward mechanisms and their effects on the participation in communities. The flexible reward mechanism developed in this thesis provides an opportunity to observe the effects of different reward strategies. For example, how different the parameter values at the start of the community are able to attract users. This information will be useful for defining the reward policies in the current and future deployments of MCComtella and other reward based communities.

Study of interactions between communities.

Previous Comtella studies were focused on a single group and its dynamics. MCComtella can be used to study both interactions within one community and interactions between communities. Studying the "travels" of users between communities will point out what factors trigger the transfer of users. The knowledge about the movement of users between communities and contributing factors for the direction of movement will be useful for both attracting and retaining users in future communities.

Study of user activity dynamics and sustainability of the community.

The study of contributions by local community members and visitors will help to appreciate the effects of collaboration between communities on their sustainability. This study can visualize activities such as sharing and rating by local and visiting community members. It would be interesting to study the effects of policy-based user modeling on the cold start problem by comparing the time taken by local and visiting users to attain the top status in the community.

Developing Libraries of Good Policies.

It would be useful to provide users with a library of policies. At a framework level, such a library will allow community owners to use successful policies of other experienced community owners. The owner will be able to add and delete policies in the library, and keep a version record of policies used over the life of the community. The owner will also be able to rollback to any past policy and can experiment with current policy after small changes. For this purpose Extensible Access Control Markup Language (XACML) can be extended for the needs of online communities. XACML provides a platform to combine rules and policies, controlling the attributes of user. XACML also offers logical and mathematical operators for user attributes. These operators can be used for calculating complex reward mechanism, thus further externalizing the policies.

According to a recent article in the Economist, the trend in online communities is towards integration, seamless transfer of users from one to another community. This thesis describes one possible approach to support such user movement though policy based user modeling, which allows users rather than community framework designers to take control of the user modeling process. Thus it aligns well with the spirit of the Participative Web, the Social Web and Web 2.0.

REFERENCES

Anderson, A. H. 2006. A comparison of two privacy policy languages: EPAL and XACML. In proceedings of the 3rd ACM Workshop on Secure Web Services (Alexandria, Virginia, USA, November 03 - 03, 2006). SWS '06. ACM Press, New York, NY, 53-60. DOI= http://doi.acm.org/10.1145/1180367.1180378

Anderson, J. R. (1988). The expert module. In M. Polson & J. Richardson (Eds.), Handbook of Intelligent Training Systems. Hillsdale, NJ: Erlbaum, 21-53.

Breslin, J. G. (2005). Towards Semantically-Interlinked Online Communities. Lecture Notes in Computer Science, vol. 3532, 500-515.

Bretzke H., Vassileva J. (2003) Motivating Cooperation in Peer to Peer Networks. Proceedings of User Modeling (UM'03), Johnstown, PA, June 22-26, Springer Verlag LNCS 2702, 2003, 218-227.

Brown, J. S., and Burton, R. R. (1978) Diagnostic models for procedural bugs in basic mathematical skills, Cognitive Science, 2, 155-191.

Bull, S. (1997) See Yourself Write: A Simple Student Model to Make Students Think. In Jameson, A., Paris, C., and Tasso, C., editors, User Modelling (Proceedings from 6th International Conference, UM 1997), 315-326.

Bull, S., and Nghiem, T. (2002) Helping Learners to Understand Themselves with a Learner Model Open to Students, Peers, and Instructors. Proceedings of Workshop on Individual and Group Modeling Methods that Help Learners Understand Themselves, International Conference on Intelligent Tutoring Systems 2002, page 5-13.

Bull, S., and Pain, H. (1995) 'Did I say what I think I said and do you agree with me?': Inspecting and Questioning the Student Model. Proceedings of World Conference on Artificial Intelligence and Education (ACCE). Charlottesville, VA, pages 501-508.

Cheng R, Vassileva J. (2005). User Motivation and Persuasion Strategy for Peer-to-Peer Communities, p., Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 7, 193-203.

Cheng R., Vassileva J. (2006) Design and Evaluation of an Adaptive Incentive Mechanism for Sustained Educational Online Communities. User Modeling and User-Adapted Interaction, special issue on User Modeling Supporting Collaboration and Online Communities, 16(3/4), 321-348.

Chin, J. P., Diehl, V. A., and Norman, K. L. 1988. Development of an instrument measuring user satisfaction of the human-computer interface. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Washington, D.C., United States, May 15 - 19, 1988). J. J. O'Hare, Ed. CHI '88. ACM Press, New York, NY, 213-218. DOI= http://doi.acm.org/10.1145/57167.57203

Cold, S. (2006). Using Really Simple Syndication (RSS) to enhance student research. SIGITE Newsl, vol. 3, 6-9, 2006.

Denaux R., V. Dimitrova, and L. Aroyo. (2004) Interactive ontology-based user modeling for personalized learning content management. In AH 2004: Workshop Proceedings Part II, pages 338–347.

Denaux, R., Aroyo, L., and Dimitrova, V. (2005). An approach for ontology-based elicitation of user models to enable personalization on the semantic web. In Special interest Tracks and Posters of the 14th international Conference on World Wide Web (Chiba, Japan, May 10 - 14, 2005). WWW '05. ACM Press, New York, NY, 1170-1171. http://doi.acm.org/10.1145/1062745.1062923

DeSouza, C. S., Preece, J. (2004). A framework for analyzing and understanding online communities. Interacting with Computers, vol. 16, 579-610.

Erik, J., Vullings D. (2005) Meta Access Management Systems. in 'Research Data Collections' Workshop. Available online at (accessed on May 2, 2009):

http://www.apac.edu.au/communication_media/RDC_workshop/Meta_access_mgt_sys-Vullings_Dalziel.pdf.

Forte A. and Bruckman A. (2008) Scaling Consensus: Increasing Decentralization in Wikipedia Governance. Available online (retrieved on May 5, 2009) at: http://www.cc.gatech.edu/~aforte/ForteBruckmanScalingConsensus.pdf

Hansen C. and McCalla, G. (2003) Active Open Learner Modelling. Proceedings of AIED2003, July 20-24, 2003, Sydney, Australia.

Harper, B., Slaughter, L., & Norman, K. (1997) Questionnaire administration via the WWW: A validation and reliability study for a user satisfaction questionnaire. In Proc. *WebNet*, 808-810

Harth A., John G. Breslin, Ina O'Murchu, Stefan Decker (2005) Linking Semantically-Enabled Online Community The Semantic Web: Research and Applications. Second European Semantic Web Conference, ESWC 2005. Proceedings (Lecture Notes in Computer Science Vol. 3532), 2005, p 500-14

Hayes-Roth, B. (1985) A blackboard architecture for control. Artificial Intelligence, 26, 251-321.

Indratmo and Vassileva J. (2005). Human and social aspects of decentralized knowledge communities, in International Web Conference 2005, 49-61.

Kagal, L., Finin, T., and Joshi, A. (2001). Trust-Based Security in Pervasive Computing Environments. Computer 34, 12 (Dec. 2001), 154-157. Available online at: http://dx.doi.org/10.1109/2.970591

Kass, R. and Finin, T. (1988). Modeling the user in natural language systems. Compute. Linguist. 14, 3 (Sep. 1988), 5-22.

Kay, J. (1995) The UM toolkit for cooperative user modelling User Modelling and User-Adapted Interaction. 4: 149-196

Kimberly, R. Swinth, Farnham, D. Shelly and Davis, P. (2003). Sharing personal information in online community member profiles. 04-13-2005. http://research.microsoft.com/scg/#papers

Kobsa, A. (1995) Supporting User Interfaces for All through User Modeling. Proceedings HCI International ’95, Yokohama. Japan. P 155-157

Kobsa, A. (2001). Generic User Modeling Systems. User Modeling and User-Adapted Interaction, 11(1), 49-.

Kobsa, A. and Fink, J. (2006). An LDAP-based User Modeling Server and its Evaluation. User Modeling and User-Adapted Interaction 16, 2 (May. 2006), 129-169. DOI= http://dx.doi.org/10.1007/s11257-006-9006-5

Lorch, M., Proctor, S., Lepro, R., Kafura, D., and Shah, S. (2003). First experiences using XACML for access control in distributed systems. In Proceedings of the 2003 ACM Workshop on XML Security (Fairfax, Virginia, October 31 - 31, 2003). XMLSEC '03. ACM Press, New York, NY, 25-37. DOI= http://doi.acm.org/10.1145/968559.968563

McCalla G., Vassileva G., Greer J., Bull, S. (2000) Active Learner Modelling, in Gautier, Frasson & VanLehn (eds.) *Proceedings of ITS'2000*, Springer LNCS 1839, 53-62.

Merrells J. (2004) XACML: XML Access Control. Available on line at: http://www.idealliance.org/papers/dx_xmle04/papers/04-01-04/04-01-04.pdf Last Visited: Oct 2006 .

Mohammed, I. and Dilts, D. M. (1994). Design for dynamic user-role-based security. Computer &. Security. 13, 9 (Oct. 1994), 661-671.

Mynatt, E. D., Adler A., Ito M., and O'Day V. L.(1997). Design for network communities. Conference on Human Factors in Computing Systems - Proceedings, 210-217, 1997. Available: http://www.sigchi.org/chi97/proceedings/paper/edm.htm

Niu X. , McCalla G. I., and J. Vassileva, (2004) Purpose-based Expert Finding in a Portfolio Management System. Computational Intelligence Journal, Vol. 20, No. 4, 548-561.

OASIS (2005) eXtensible Access Control Markup Language 2 (XACML) Version 2.0 OASIS Standard, 1 Feb 2005 http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf . Last visited: Oct 2006

Park, J. S., Sandhu, R., and Ahn, G. (2001). Role-based access control on the web. ACM Trans. Inf. Syst. Secur. 4, 1 (Feb. 2001), 37-71. http://doi.acm.org/10.1145/383775.383777

Preece, J. D. Maloney-Krichmar and Abras C. (2003) History of Online Communities. Encyclopaedia of Community: From Village to Virtual World. Thousand Oaks: Sage Publications, 1023-1027.

Preece, J. (2000). Online Communities: Designing Usability, Supporting Sociability. Wiley: Chichester..

Rich, E. (1979) User modeling via Stereotypes. Cognitive Science, 3: pp 329-354.

Rich, E.: (1983) Users are Individuals: Individualizing User Models. International Journal of Man-Machine Studies 18(3): 199-214

Sandhu, R. and Park, J. S. (1998). Decentralized user-role assignment for Web-based intranets. In Proceedings of the Third ACM Workshop on Role-Based Access Control Fairfax, Virginia, USA, October 1998. Available on line at http://doi.acm.org/10.1145/286884.286887

Schlungbaum E. & Elwert T. (1996) Automatic user interface generation from declarative models, Proceedings CADUI'96, Naumur, Belgium. (Presentation available at: www.isys.ucl.ac.be/bchi/cadui/96/files96/Schlungbaum-CADUI96.ppt*)*

Self, J. (1999). Open Sesame?: Fifteen variations on the theme of openness in learning environments, International Journal of AI in Education 10, 1020-1029.

Seth P. (2004). Sun's XACML Implementation. Programmer's guide for version 1.2. 2006 (Aril, 2006).

Sun L., Vassileva J. (2006) Social Visualization Encouraging Participation in Online Communities, In Groupware: Design, Implementation, and Use, Proceedings of CRIWG'2006, Springer LNCS 4154, 349-363.

Vassileva J., McCalla G., Greer J. (2003) Multi-Agent Multi-User Modeling, User Modelling and User Adapted Interaction 13:(1), 2003, pp. 1-31.

Vassileva, J. (2002) Motivating Participation in Peer to Peer Communities, Proceedings of the Workshop on Emergent Societies in the Agent World, ESAW'02, Madrid, 16-17 September, 2002, Springer Verlag LNAI 2577, 141-155

Vassileva, J., Greer, J., McCalla, G. (1999) Openness and Disclosure in Multi-Agent Learner Models. Workshop on Open, Interactive, and Other Overt Approaches to Learner Modeling (Proceedings from 9th International Conference, AIED 1999), pages 43-49.

Appendix A

This appendix presents the sheet that observer used to record the performance of a participant through the five scenarios.

Observer Report

In this experiment user’s performed different task in a scenario based study. The Observer provided a brief introduction of Comtella. He gave a walkthrough presentation of different Comtella screens. After the presentation user was provided with an opportunity to go through different screens and navigate Comtella web page menu. Finally Observer asked users if the have any question and confusion regarding the Comtella.

After this introductory session Observer started handing over the tasks one by one. Observer was sitting with the user to monitor the accuracy of the task and recorded each task on observe sheet with following table.

User performed the task?

□ User could not perform the task □User performed task with your help □User perform the task with explanation □User performed the task smoothly

User could not perform the task: When user could not complete the task.

User performed task with your help: When user performed the task with intervention of the observer. The Observer clicked mouse or used key board to help user to complete the task.

User performed the task with explanation: When user asked for help and observer explained him what is the next step. User never used keyboard, mouse or pointed anything on screen.

User performed the task smoothly: When User perform the task without any interaction with the observer.

After performing each task user was given a questionnaire with question about the task being performed. During this period observer was away from the users and had no interaction at all.

Appendix B

This appendix lists the questions asked to participants after each task in the 5 scenarios

1. **Do you have an account or accounts on any of the following websites? Check those that you have personally used and are familiar with.**

__ Google Group

__ Yahoo Groups

__ My Space

__ Windows Live Spaces

__ Hotmail.com

__ GMail

__ YahooMail

__ Digg

__ Pageflakes

__ groups@AOL

__ GroupSense : Online Collaboration

__ Comtella

2. Scenario: Create a Community

Please create a community. Give it a name, description, and add some keywords.

Scenario: Create a community Observer sheet

User performed the task?

□ User could not perform the task □User performed task with your help □User perform the task with explanation □User performed the task smoothly

Scenario Create a Community

2.1 Do you think that giving the option of "creating new community" will help the overall sustainability of communities?

□ Not at all helpful □ May be helpful □ Helpful □ Extremely helpful □ Don't know

2.2 I found that the number of keyword entries for my new community was...

□ Not at all appropriate □ May be appropriate □ Appropriate □ Very appropriate

2.3 Please rank the Create community screen of Comtella on a 5-point scale (-2: very poor; +2 very good)

-2 Very Poor -1 Poor 0 Satisfactory +1 good +2 Very good

		-2	**-1**	**0**	**+1**	**+2**
2.3.1	Overall					
2.3.2	Usability					
2.3.3	Reliability (crashes etc.)					
2.3.4	Menu					

3. Scenario creating a *“Status Policy”* for new Community

Please set following for *“Status Policy”* of your community.

- Weight for rating quality 3
- Weight for Paper quality 3

Add the following rules to the *“Reward Policy”* for user actions.

Add new share	20%
Give remarks	10%
Rating	20%
Upload file	20%
Post message	20%
Add new community	10%

Set the following rules in the *“Status Policy”* of your community.

Status	Start	End
Plastic	0	59
Bronze	60	89
Silver	90	119
Gold	120	500

Set the *“Access Policy”* of your community to following values:

Status	Access to actions
Plastic	Allow Post Message, Rate Sharing
Bronze	Allow Share Link, Post Message, Rate Sharing
Silver	Allow Share Link, Share File, Post Message, Rate Sharing
Gold	Allow Share Link, Share File, Post Message, Rate Sharing

Scenario: Creating a *“Status Policy”* of the new community Observer sheet

User performed the task?

□ User could not perform the task □User performed task with your help □User perform the task with explanation □User performed the task smoothly

Scenario: Creating a *“Status Policy”* of the new community

3.1 The community owner should be able to set reward rules for his/her community members.

□ Do not agree □ Not agree □ Somewhat agree □Agree □Strongly agree

3.2 Would you support the idea of limiting users’ actions in the community based on their status (for example limiting who can read, share and comment based on their status)

□ Do not support at all □ Do not support □Somewhat support □ Support □ Fully support

3.3 How important is the visualization of your status in a community.

□ Not important at all □ Not important □ May be important □ Important □ Extremely important

3.4 Do you agree with the statement?, “Communities may have their own set of reward rules for participation”

□ Do not agree at all □ Not agree □ Somewhat agree □Agree □ Strongly agree

3.5 The Comtella *policy editing screen* offers a set of actions (Add new share, Give remarks, Rating, Upload file, Post message, Add new community) for reward rules in the *Status Policy*, is this set

□ Not at all appropriate □ Not appropriate □ Some what appropriate □Appropriate □Very appropriate

3.6 The Comtella *policy editing screen* offers a default status policy with a set of status rules. Do you think these rules are appropriate for most communities?

□ Not at all appropriate □ Not appropriate □ Some what appropriate □Appropriate □Very appropriate

3.7 The Comtella *policy editing screen* allows users to set certain access rights (share link/file, post message and rating). Do you think these rights are appropriate for the *Status Policy*?

□ Not at all appropriate □ Not appropriate □ Some what appropriate □Appropriate □Very appropriate

3.8 If you have selected “Not at all appropriate” or “some what appropriate” in previous question. Please suggest additional access rights that should be included or removed in the *Status Policy*.

Included Removed

3.9 I found that number of status options (10) for my new community was...

□ Not at all appropriate □ Not appropriate □ Some what appropriate □Appropriate □Very appropriate

3.10 Please rank the Status policy editing screen of Comtella on a 5-point scale (-2: very poor; +2 very good):

-2 Very Poor -1 Poor 0 Satisfactory +1 good +2 Very good

	-2	**-1**	**0**	**+1**	**+2**
Overall					
Usability					
Reliability (crashes etc.)					
Policy file upload					
Adding and removing Action					
Over all readability of policy					
Readability of XML file					

4. Scenario: Edit the “Status Policy” of the community

Please, by using the “edit policy” option, edit the *Status Policy* of your community.

Remove the current *Post Message* and *Add New Community* actions and set them to the following:

Post Message	10%
Add New Community	20%

Reset the *Status Policy* rules of your community following:

Status	Start	End
Plastic	0	69
Bronze	70	99
Silver	100	199
Gold	200	500

Allow the *Plastic* status to use *Rate Sharing* in the *Access Policy*.

Scenario: Edit the *“Status Policy” of* the community
Observer sheet

User performed the task?

□ User could not perform the task □User performed task with your help □User perform the task with explanation □User performed the task smoothly

Scenario: Edit the *“Status Policy” of* the community

4.1 Would you prefer to edit the policy document in some text editor and upload with *Edit policy* option?

□ Not at all □ No □ May be yes □ Yes □ Definitely yes

4.2 If you found a policy of some other community suitable for your community. Would you like to request the XML based policy document from that community owner?

□ Not at all □ No □ May be yes □ Yes □ Definitely yes

5. Scenario: Create a *“Transfer Policy”* of the community

Please create a transfer policy from the “Gardening Community” to your community. And perform following tasks:

Task 1

Allow visiting users to use the *Status Policy* of “Gardening Community” as a *Transfer Policy*.

Task 2

Allow visiting users to use the *Status Policy* of your community as a *Transfer Policy*.

Task 3

Create a new *Transfer Policy* from the “Gardening Community” to your community.

Scenario: Create a *“Transfer Policy”* of the community

Observer sheet

User performed the task?

□ User could not perform the task □User performed task with your help □User perform the task with explanation □User performed the task smoothly

Scenario: Create a *“Transfer Policy”* of the community

5.1 When a user moves from one community to another. Do you think his/her status may be transferred to another community?

□ Not at all agree □ Not agree □ Somewhat agree □Agree □Strongly agree

5.2 Do you think the community owner should be able to set rules for transfer of user from one community to another?

□ Not at all agree □ Not agree □ Somewhat agree □Agree □Strongly agree

5.3 Please rank the Transfer policy editing screen of Comtella on a 5-point scale (-2: very poor; +2 very good):
-2 Very Poor -1 Poor 0 Satisfactory +1 good +2 Very good

	-2	**-1**	**0**	**+1**	**+2**
5.3.1 Overall					
5.3.2 Usability					
5.3.3 Reliability (crashes etc.)					
5.3.4 Over all readability of policy					

Scenario: Create a *“Transfer Policy”* of the community
Observer sheet

Task 1

User performed the task 1?

□ User could not perform the task

□User performed task with your help

□User perform the task with explanation

□User performed the task with two mouse clicks

Task 2

User performed the task 2?

□ User could not perform the task

□User performed task with your help

□User performed the task with explanation

□ User performed the task with one mouse clicks

Task 3

User performed the task 3?

□ User could not perform the task

□User performed the task with your help

□ User performed the task with explanation

□User performed the task smoothly

6. Scenario: Edit a “*Role Policy*” of the community

Please select the option from menu to edit a role policy of your community. Perform following tasks:

Allow the *Operator* role to *Delete Link*

Revoke the right of the *Expert* to *Edit Policy*

Revoke the right of the *Expert* to *Edit Role*

Add a new role called *Friend* and assign it the rights of an *Owner*.

Assign the role of *Operator* to the user *“Foo”*

Scenario: Edit a “*Role Policy*” of the community

Observer sheet

User performed the task?

□ User could not perform the task □User performed task with your help □User perform the task with explanation □User performed the task smoothly

Scenario: Edit a “*Role Policy*” of the community

6.1 As community owner, would you like to delegate the roles of community administration to other members?

□ Not at all □ No □ May be yes □Yes □Definitely yes

6.2 Will you use power delegations as strategy to attract more users in your community?

□ Not at all □ No □ May be yes □Yes □Definitely yes

6.3 The Comtella “*Role Policy Screen*” offers a default *Role Policy*. Do you think this default *Role Policy* is appropriate for most of the communities?

□ Not at all appropriate □ Not appropriate □ Some what appropriate □Appropriate □Very appropriate

6.4 The Comtella “*Role Policy Screen*” allows users to set certain access rights (*Delete Link, Create Community and Edit Link/Policy/Role*). Do you think these rights are appropriate for the *Role Policy*?

□ Not at all appropriate □ Not appropriate □ Some what appropriate □Appropriate □Very appropriate

6.5 If you have selected “Not at all appropriate” or “some what appropriate” in the previous question. Please suggest additional access rights that should be included or removed in *Role Policy.*

Included	Removed

6.6 I found that number of role options (10) for my new community was...

☐ Not at all appropriate ☐ Not appropriate ☐ Some what appropriate ☐ Appropriate ☐ Very appropriate

6.7 Would you prefer to edit the *Role Policy* document in a text editor and upload it with "*Role Policy Screen*"?

☐ Not at all ☐ No ☐ May be yes ☐ Yes ☐ Definitely yes

Scenario: Edit a “*Role Policy*” of the community

6.8 If you found a *Role Policy* of some other community appropriate for your community, would you like to request the XML based role policy document from that community owner?

□ Not at all □ No □ May be yes □ Yes □ Definitely yes

6.9 Please rank the role policy editing screen of Comtella on a 5-point scale (-2: very poor; +2 very good)

-2 Very Poor -1 Poor 0 Satisfactory +1 good +2 Very good

	-2	**-1**	**0**	**+1**	**+2**
Overall					
Usability					
Reliability (crashes etc.)					
Policy file upload					
Add and removing roles					
Over all readability of policy					
Read ability of XML file					

7. As user of the Comtella

7.1 If you see that a certain action is rewarded by a status policy, will you perform that action more frequently to get a higher status?

□ Not at all □ No □ May be yes □ Yes □ Definitely yes

7.2 How important is it to have your status transferred when you move from one community to another?

□ Not important at all □ Not important □ May be important □ Important □ Extremely important

7.3 Does the status of a user affect your rating of his/her contributions?

□ No affect at all □ Not important □ May affect □ Affect □ Definitely affected

7.4 Does a higher status in a community motivate you to participate more?

□ Not motivate all □ Not motivate □ May be motivate □ Motivate □ Fully motivate

7.5 Will a better status in a community motivate you to invite other people to join that community?

□ Not motivate at all □ Not motivate □ May be motivate □ Motivate □ Fully motivate

7.6 Will having some administrative role in a community motivate you to participate more?

□ Not motivate all □ Not motivate □ May be motivate □ Motivate □ Fully motivate

7.7 Will having some administrative role in a community motivate you to invite other people to join the community?

□ Not motivate all □ Not motivate □ May be motivate □ Motivate □ Fully motivate

7.8 Which of the following statement do you support?

□ Status should be transferable □ Status should only be earned by time

Please describe why?

7.9 Will you be comfortable if someone moves from another community/group and gets a higher status based on his/her status in the previous community?

□ Not at all □ No □ May be yes □ Yes □ Definitely yes

Please describe why?

7.10 Which of the following factors prevents you from joining the new community/Group (check all applicable)?

□ Filling up new signup form

□ Starting from scratch to earn reputation

□ Transferring files/blog and data

□ Other please specify

□ Setting new preferences

□ Privacy concerns

□ working with new people

□ Learning new skills

7.11 Which of the following are important for you as reward mechanisms?

□ Different color and layout on Screen

□ More functionality and access rights

□More Storage rights

□ Reward point/ Status to be distinct among community members

□ Other please specify

Appendix C

This appendix lists the question asked to participants about the usability and overall reaction about the Comtella screens and system capabilities.

Identification number: ______________________

Age: ______

Gender: ____ **male**

____ **female**

1. PART 1: Overall User Reactions

Please circle the numbers which most appropriately reflect your impressions about using this computer system.

Not Applicable = NA.

1. Overall reactions to the system:	terrible	wonderful	1 2 3 4 5 6 7 8 9	NA
1.2	frustrating	satisfying	1 2 3 4 5 6 7 8 9	NA
1.3	dull	stimulating	1 2 3 4 5 6 7 8 9	NA
1.4	difficult	easy	1 2 3 4 5 6 7 8 9	NA
1.5	inadequate power	adequate power	1 2 3 4 5 6 7 8 9	NA
1.6	rigid	flexible	1 2 3 4 5 6 7 8 9	NA

2. PART 2: Screen

2.1 Screen layouts were helpful never always
1 2 3 4 5 6 7 8 9 NA

2.1.1 Amount of information that can be
displayed on screen inadequate adequate
1 2 3 4 5 6 7 8 9 NA

2.1.2 Arrangement of information on screen illogical logical
1 2 3 4 5 6 7 8 9 NA

2.2 Sequence of screens confusing clear
1 2 3 4 5 6 7 8 9 NA

2.2.1 Next screen in a sequence unpredictable predictable
1 2 3 4 5 6 7 8 9 NA

2.2.2 Going back to the previous screen impossible easy
1 2 3 4 5 6 7 8 9 NA

2.2.3 Progression of work related tasks confusing clearly marked
1 2 3 4 5 6 7 8 9 NA

Please write your comments about the screens here:

____

____

____

____

3. PART 3: Learning

3.1 Learning to operate the system difficult easy
1 2 3 4 5 6 7 8 9 NA

3.1.1 Getting started difficult easy
1 2 3 4 5 6 7 8 9 NA

3.1.2 Learning advanced features difficult easy
1 2 3 4 5 6 7 8 9 NA

3.1.3 Time to learn to use the system slow fast
1 2 3 4 5 6 7 8 9 NA

3.2 Remembering names and use of commands difficult easy
1 2 3 4 5 6 7 8 9 NA

3.2.1 Remembering specific rules about
entering commands difficult easy

1 2 3 4 5 6 7 8 9 NA

3.3 Tasks can be performed in a straight-forward manner

never always

1 2 3 4 5 6 7 8 9 NA

3.3.1 Number of steps per task

too many just right

1 2 3 4 5 6 7 8 9 NA

3.3.2 Steps to complete a task follow a logical sequence

never always

1 2 3 4 5 6 7 8 9 NA

3.3.3 Feedback on the completion of of steps

unclear clear

1 2 3 4 5 6 7 8 9 NA

Please write your comments about learning here:

____

____

____

____

4. PART 4: System Capabilities

4.1 System speed too slow fast enough
1 2 3 4 5 6 7 8 9 NA

4.1.1 Response time for most operations too slow fast enough
1 2 3 4 5 6 7 8 9 NA

4.1.2 Rate information is displayed too slow fast enough
1 2 3 4 5 6 7 8 9 NA

4.2 The system is reliable never always
1 2 3 4 5 6 7 8 9 NA

4.2.1 Operations are undependable dependable
1 2 3 4 5 6 7 8 9 NA

4.2.2 System failures occur frequently seldom
1 2 3 4 5 6 7 8 9 NA

Please write your comments about system capabilities here:

____

____

____

Appendix D

This appendix list the user responses for questionnaire given in appendix C

Please refer Appendix C for the description of questionnaire.

Participants	Overall user reactions						
	1.1	**1.2**	**1.3**	**1.4**	**1.5**	**1.6**	**Average**
1	8	8	5	8	8	4	6.833333
2	8	8	7	8	9	9	8.166667
3	7	9	5	7	8	7	7.166667
4	7	6	3	8	8	6	6.333333
5	9	9	9	8	8	8	8.6
6	9	8	9	8	9	8	8.5
7	8	7	7	9	8	8	7.833333
8	7	7	6	7	7	8	7
9	7	8	7	8	9	8	7.833333
10	7	7	8	9	8	7	7.666667
11	8	7	7	8	9	8	7.833333
12	7	7	7	8	8	8	7.5
							7.605556

Table C-1 : Overall user reaction about Comtella System (Ref. Appendix C)

Participants	Screen							
	2.1	**2.1.1**	**2.1.2**	**2.2**	**2.2.1**	**2.2.2**	**2.2.3**	**Average**
1	8	8	7	8	8	8	8	7.857143
2	7	8	7	7	6	8	8	7.285714
3	9	9	9	9	1	6	7	7.142857
4	9	7	8	8	6	9	6	7.571429
5	9	9	9	9	7	9	9	8.714286
6	9	8	8	9	9	9	9	8.714286
7	7	8	7	6	5	3	8	6.285714
8	6	6	7	7	5	4	5	5.714286
9	7	7	5	5	8	8	8	6.857143
10	9	8	8	8	8	7	9	8.142857
11	6	8	6	8	6	8	8	7.142857
12	9	8	7	8	8	8	8	8
								7.452381

Table C-2 : Overall user reaction about Comtella screens (layouts, sequence of screens) (Ref. Appendix C)

Participants	Learning										
	3.1	3.1.1	3.1.2	3.1.3	3.2	3.2.1	3.3	3.3.1	3.3.2	3.3.3	Average
1	8	8	8	7	8	8	8	8	8	8	7.875
2	9	7	7	7	7	7	9	8	9	9	7.625
3	8	8	5	4	5	9	6	7	1	9	6.5
4	8	7	8	6	8	9	9	8	8	7	7.875
5	9	9	8	9	7	8	9	9	9	9	8.5
6	9	9	8	8	8	9	9	9	9	9	8.625
7	7	8	5	7	8	6	8	8	8	8	7.125
8	6	6	7	6	6	7	6	4	7	6	6
9	9	8	9	9	9	9	8	8	8	9	8.625
10	9	9	9	9	9	9	9	9	9	8	9
11	7	8	5	8	6	6	6	6	8	9	6.5
12	7	6	7	6	8	8	8	8	8	9	7.25
											7.625

Table C-3 : Overall user reaction about learning (getting started, advance features, remembering name, use of commands and simplicity of task) use of Comtella (Ref. Appendix C)

Participants	System Capabilities						
	4.1	4.1.1	4.1.2	4.2	4.2.1	4.2.2	Average
1	8	8	8	6	7	8	8.833333
2	7	8	9	9	8	9	8.333333
3	7	7	9	8	8	8	7.833333
4	8	9	6	9	8	9	8.166667
5	8	7	9	9	9	9	8.5
6	8	9	9	9	9	9	8.833333
7	7	7	8	8	8	9	7.833333
8	6	5	6	6	5	8	6
9	8	8	8	8	9	9	8.333333
10	7	9	9	9	9	8	8.5
11	9	9	8	9	9	9	8.833333
12	9	9	9	9	7	9	8.666667
							8.222222

Table C-4 : Overall user reaction about the capabilities (system speed and reliability) of Comtella (Ref. Appendix C)

Overall communities membership

1.1 Do you have an account or accounts with any of the following communities/websites? Check those that you have personally used and are familiar with.

S.No	Google Group	Yahoo Group	My Space	Windows Live Space	Hotmail	Gmail	Yahoo Mail	Digg	Page flakes	Group@AOL	Groups: Online Collaboration	Comtella	I-Help	Total
1		1			1	1	1					1		5
2	1	1			1	1	1						1	6
3					1		1							2
4				1	1	1								3
5			1		1		1							3
6	1			1	1	1	1					1	1	7
7	1	1			1	1	1							5
8			1	1	1		1							4
9		1			1	1	1							4
10	1	1		1	1	1	1						1	7
11	1	1	1		1	1	1							6
12		1			1	1	1							4
Total	5	7	3	4	12	9	11	0	0	0	0	2	3	
													Average	4.727273

Table D-1.1 : A survey of user memberships on other websites

Overall communities membership

1.2 Which of the following factors prevent you from joining new communities?(check all applicable)

S. No	Filling up new sign form	Setting new preferen-ces	Working with new people	Starting from scratch to earn reputation	Privacy concerns	Lear-ning new skills	Transferring files/blog and data	Transfer-ring Contacts
1	1	1		1			1	1
2	1			1	1			
3	1	1						1
4	1				1			
5	1							1
6					1		1	
7	1				1		1	
8	1		1	1			1	1
9			1					
10				1	1		1	
11		1		1	1	1		1
12	1							
Total	8	3	2	5	6	1	5	5

Table D-1.2 : Factors preventing users to join a new community

Overall communities membership

1.3 Will having some special role in a community motivate you to participate more?

S. No.	Not Motivate at all	Not Motivate	May Motivate	Motivate	Strongly motivate
1				1	
2				1	
3			1		
4	1				
5	1				
6				1	
7			1		
8			1		
9			1		
10			1		
11				1	
12				1	
Total	2	0	5	5	0
	16.66667%	0%	41.66667%	41.66667%	0

Table D-1.3 : Impact of special roles in community over user's participation

Overall communities membership					
1.4 Will having some special role in a community motivate you to invite other people to join the community?					
S. No.	Not Motivate at all	Not Motivate	May Motivate	Motivate	Strongly motivate
1				1	
2					1
3		1			
4		1	1		
5			1		
6					
7			1		
8				1	
9			1		
10			1		
11			1		
12				1	
Total	0	2	6	3	1
	0	16.66667%	50%	25%	8.333333%

Table D-1.4: Impact of special roles in community on peers interaction.

Overall communities membership

1.5 Which of the following statements do you support?

S. No.	Status should be transferable	Status should apply only to the community where it was earned	Comments
1	1		So I don't start from scratch, provided that both communities current and new) are at the same level of proficiency/rating system.
2		1	I think it should apply only to the the community where it was earned because different community has different rules to earn the status. The rules of one community may not apply to the other one
3	1		To let others know who are expert
4	1		Status refers to who you are as individual, not some online alias
5		1	The status you earned in one community is based on what you contributed to this community. It is possible that what you did may not be useful or helpful for other communities.
6	1		Save your time to start over again
7	1		
8	1		Sharing stuff is good for communication
9	1		Because you worked hard to earn it in one community, its not fair to start from scratch in another one
10		1	For privacy concern
11		1	Each community caters different demographics and may have a different objectives
12	1		to reflect the participation level in previous group
Total	8	4	
	66.66667%	33.33333%	

Table D-1.5: Transferable User status?

Overall communities membership

1.6 Will you be comfortable if someone moves to your community from another community/group and gets a higher status based on his/her status in the previous community?

S. No.	Not at all	NO	May be yes	Yes	Definitely yes	Comments
1		1				If I get treated the same way I move to his/her community then I believe that's fair, provided again that both communities are at the same level. It has to work both ways
2		1				I believe, every community has different methods to assign points based on which the status is determined) so a person who is new to a community, can not earn more points then a person who has already been working on it for a while……..
3			1			it would let me know who are other experts etc.
4			1			The reputation accumulated from previous experiences is applicable to different environments
5		1				It is not fair to other people in our community
6			1			If it's similar community. Its ok.
7			1			
8			1			Knowing people is most important . I am not so care about the status
9				1		I guess these should be factors that determine his status in my community, these factors should be fair. For example , if a person transfer from university to study in another may be not all courses he took in the old will count in the new one, but some of them will.
10			1			depends on where he/she move from, if she from a well accepted community then probably there is no problem
11		1				
12		1				could be a person with unique abilities which may help the group objectives and goals
Total	0	5	6	1	0	
	0%	41.66667%	50%	8.333333%	0	

Table D-1.6: Response about new comer in community.

Overall communities membership

1.7 Which of the following are important for you as reward mechanisms?

S. No	Different color and layout on screen	More functionality and access rights	More storage	Reward Points	Total	Comments
1		1	1		2	May be a higher status name
2		1	1	1	3	
3		1	1		2	
4		1	1		2	
5		1			1	
6	1	1	1	1	4	
7	1	1	1		3	
8			1	1	2	
9	1	1	1	1	4	
10		1	1		2	
11		1	1		2	
12	1	1			2	
Total	4	11	10	4		
	33.33333%	91.66667%	83.33333%	33.33333%		

Table D-1.7: Reward mechanism.

Task Error

SNO	Task 1				Task 2				Task 3				Task 4.1				Task 4.2				Task 4.3				Task 5			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1				1				1				1				1				1				1				1
2				1				1				1				1				1				1				1
3				1				1			1				1					1				1				1
4				1			1					1				1				1				1				1
5				1			1					1			1					1				1				1
6				1				1				1				1				1				1				1
7				1				1			1				1					1				1				1
8				1			1				1				1			1						1				1
9				1			1				1				1					1				1			1	
10				1				1				1		1					1				1					1
11				1				1			1					1				1				1				1
12				1				1				1				1			1					1		1		
	0	0	0	12	0	0	4	8	0	0	5	7	0	1	5	6	0	1	2	9	0	0	1	11	0	1	1	10
	0	0	0	100	0	0	33.33%	66.66%	0	0	41.66%	58.33%	0%	8.33%	41.67%	50%	0%	8.33%	16.66%	75%	0%	0%	8.33%	91.66%	0%	8.33%	8.33%	83.33%

Table D-1.8 Task Error Record from observer's sheet
1-Could not perform , 2-Performed with help , 3-Performed with explanation, 4-Performed (Ref. Appendix A for description of these categories)

Scenario: Create a Community

2.1 Do you think that giving the option to “Create new community” is useful?

S. No.	Not at all user full	Not useful	May be user full	Useful	Extremely user full	Don't know
1				1		
2					1	
3					1	
4				1		
5					1	
6				1		
7					1	
8				1		
9					1	
10				1		
11					1	
12				1		
Total	0	0		6	6	0
	0.00	0.00	0.00	50.00	50.00	

Table D-2.1 Create Community

Scenario: Create a Community

2.2 I found that the number of keyword entries (10) for my new community was...

S. No.	Not at all appropriate	Not appropriate	May be appropriate	appropriate	Very appropriate
1				1	
2					1
3			1		
4				1	
5				1	
6				1	
7					1
8			1		
9					1
10			1		
11				1	
12					1
Total	0	0	3	5	4
	0.00	0.00	25.00	41.67	33.33

Table D-2.2 Number of keywords in Create Community Screen

Scenario: Create a Community

2.4 Please rank the “Create community screen" of Comtella on a 5-point scale (-2: very poor; +2 very good)

	Over All					Usability					Reliability (crashes etc.)					Menu				
S. No	Very poor -2	Poor -1	Satisfactory 0	Good 1	Very Good 2	Very poor -2	Poor -1	Satisfactory 0	Good 1	Very Good 2	Very poor -2	Poor -1	Satisfactory 0	Good 1	Very Good 2	Very poor -2	Poor -1	Satisfactory 0	Good 1	Very Good 2
1					1					1					1					1
2				1						1					1					1
3					1					1					1				1	
4				1						1					1				1	
5					1					1					1					1
6					1					1					1					1
7					1					1				1						1
8				1						1				1					1	
9					1					1					1					1
10				1						1					1				1	
11					1					1					1				1	
12				1					1					1					1	
	0	0	0	5	7	0	0	0	1	11	0	0	0	3	9	0	0	0	6	6
	0.00	0.00	0.00	41.67	58.33	0.00	0.00	0.00	8.33	91.67	0.00	0.00	0.00	25.00	75.00	0.00	0.00	0.00	50.00	50.00

Table D-2.3 Overall Ranking of “Create community screen"

Scenario: Creating a "Status Policy" of the new community

3.1 The community owner should be able to set reward rules for his/her community members.

S. No.	Do not agree at all	Not agree	Somewhat agree	Agree	Strongly agree	Don't know	
1					1		
2					1		
3					1		
4					1		
5					1		
6					1		
7					1		
8					1		
9					1		
10					1		
11					1		
12				1			
Total	0	0		1	11	0	
	0.00	0.00	0.00	8.33	91.67	0.00	0.00

Table D-3.1 Creating Status policy

Scenario: Creating a "Status Policy" of the new community

3.2 Would you support the idea of limiting users' actions in the community based on their status (for example limiting who can read, share and comment based on their status)?

S. No.	Do not Support at all	Do not support	Somewhat support	Support	Fully Support
1				1	
2					1
3				1	
4				1	
5				1	
6					1
7				1	
8				1	
9				1	
10					1
11				1	
12				1	
Total	0	0	0	9	3
	0.00	0.00	0.00	75.00	25.00

Table D-3.2 Limiting user's action based on status.

Scenario: Creating a "Status Policy" of the new community

3.3 Do you agree with the statement, “Communities may have their own set of reward rules for participation"? (Based on their status)?

S. No.	Do not agree at all	Not agree	Somewhat agree	Agree	Strongly agree
1		1			
2					1
3					1
4					1
5				1	
6				1	
7					1
8				1	
9				1	
10				1	
11				1	
12				1	
Total	0	1	0	7	4
	0.00	8.33	0.00	58.33	33.33

Table D-3.3 Community and reward rules.

Scenario: Creating a "Status Policy" of the new community

3.4 How important is the visualization of your status in a community?

S. No.	Not at all important	Not important	May be important	Important	Extremely important
1					1
2					1
3				1	
4			1		
5			1		
6					1
7				1	
8				1	
9					1
10				1	
11			1		
12				1	
Total	0	0	3	5	4
	0.00	0.00	25.00	41.67	33.33

Table D-3.4 Visualization of Status.

Scenario: Creating a "Status Policy" of the new community					
3.5 Do you understand the default status policy described in the policy editing screen?					
S. No.	Not at all understand	Not understand	May be understand	understand	Clearly understand
1					1
2				1	
3				1	
4				1	
5				1	
6				1	
7				1	
8				1	
9				1	
10					1
11				1	
12				1	
Total	0	0	0	10	2
	0.00	0.00	0.00	83.33	16.67

Table D-3.5 Understanding the Status policy.

Scenario: Creating a "Status Policy" of the new community	
3.6 The Comtella "policy editing screen" offers a default Status Policy. Please, describe in your word what rights the Status Policy assign.	
1	Gold can: share link…
2	The rights assigned are …..
3	The status policy offers the owner ….
4	ability to share files among community members*******
5	The status in community determines the access rights ……
6	Gold and silver ……..
7	Status policy *******
8	Status policy should assign the rights for visiting the level of user and rewards, share all rights to golden member, and with the level down access right go down
9	Share link, share file, post message, rate sharing
10	Limitation of four action share, post message i.e. the work of sharing or editing will increase the reward……
11	Rights are allowing members to share links and files with other community members, post messages in the community members to share ratings based on status of members in the community.
12	Add new community add new share, give remarks, post messages rating, upload file add policy, edit community, edit link, edit policy, logout, search **********

Table D-3.6 Description of the Status policy.

Scenario: Creating a "Status Policy" of the new community

3.7 The Comtella policy editing screen offers a set of actions (Add new share, Give remarks, Rating, Upload file, Post message, Add new community) for reward rules in the Status Policy. Is this set :

S. No.	Not at all appropriate	Not appropriate	Some what appropriate	Appropriate	Very appropriate
1				1	
2				1	
3					1
4				1	
5				1	
6				1	
7				1	
8				1	
9				1	
10				1	
11			1		
12				1	
Total	0	0	1	10	1
	0.00	0.00	8.33	83.33	8.33

Table D-3.7 Set of actions.

Scenario: Creating a "Status Policy" of the new community

3.8 If you have selected “Not at all appropriate” or “Not appropriate”, please specify actions you deem appropriate for the Status Policy.

1	none
2	none
3	none
4	none
5	none
6	none
7	none
8	none
9	none
10	none
11	none
12	none

Table D-3.8 Suggest new actions

Scenario: Creating a "Status Policy" of the new community

3.9 The Comtella policy editing screen offers a default status policy with a set of status rules. Do you think this default status policy is appropriate for most communities?

S. No.	Not at all appropriate	Not appropriate	Some what appropriate	Appropriate	Very appropriate
1			1		
2			1		
3				1	
4			1		
5					1
6				1	
7				1	
8				1	
9				1	
10					1
11				1	
12				1	
Total	0	0	3	7	2
	0.00	0.00	25.00	58.33	16.67

Table D-3.8 Default Policy

Scenario: Creating a "Status Policy" of the new community

3.10 If you have selected “Not at all appropriate” or “Not appropriate”, please specify some alternatives rules that you deem appropriate for the Status Policy.

1	none
2	none
3	none
4	none
5	none
6	none
7	none
8	none
9	none
10	none
11	none
12	none

Table D-3.10 Suggest new actions

Scenario: Creating a "Status Policy" of the new community

3.11 The Comtella policy editing screen allows users to set certain access rights (share link/file, post message and rating). Do you think these rights are appropriate for the Status Policy?

S. No.	Not at all appropriate	Not appropriate	Somewhat appropriate	appropriate	Very appropriate
1			1		
2				1	
3				1	
4				1	
5					1
6				1	
7				1	
8				1	
9			1		
10				1	
11				1	
12				1	
Total	0	0	2	9	1
	0	0	16.66667	75	8.333333

Table D-3.11 List of rights.

Scenario: Creating a "Status Policy" of the new community

3.12 If you have selected “Not at all appropriate” or “Not appropriate” in the previous question, please suggest additional access rights that should be included or removed from the Status Policy.

1	none
2	none
3	none
4	none
5	none
6	none
7	none
8	none
9	none
10	none
11	none
12	none

Table D-3.12 Suggest new user rights

Scenario: Creating a "Status Policy" of the new community

3.13 I found that number of status options (10) for my new community was...appropriate for the Status Policy?

S. No.	Not at all appropriate	Not appropriate	Somewhat appropriate	Appropriate	Very appropriate
1				1	
2				1	
3			1		
4				1	
5				1	
6				1	
7				1	
8				1	
9				1	
10				1	
11				1	
12				1	
Total	0	0	1	11	0
	0.00	0.00	8.33	91.67	0.00

Table D-3.13 Status option.

Scenario: Editing a "Status Policy" of a community

4.1 Would you prefer to edit the policy document in some text editor and upload with Edit policy option?

S. No.	Not at all	No	May be yes	Yes	Definitely yes
1		1			
2		1			
3		1			
4	1				
5		1			
6					1
7			1		
8				1	
9		1			
10				1	
11		1			
12	1				
Total	2	6	1	2	1
	16.67	50.00	8.33	16.67	8.33

Table D-4.1 Edit Status Policy.

Scenario: Editing a "Status Policy" of a community

4.2 If you find a policy of some other community suitable for your community, would you request the XML based policy document from that community owner?

S. No.	Not at all	No	May be yes	Yes	Definitely yes
1				1	
2				1	
3		1			
4				1	
5				1	
6				1	
7				1	
8				1	
9					1
10				1	
11				1	
12				1	
Total	0	1		10	1
	0.00	8.33	0.00	83.33	8.33

Table D-4.1 Edit Status Policy in XML.

Scenario: Editing a "Status Policy" of a community

4.3 Do you have any comments or suggestions regarding editing existing policies?

I liked the idea of requesting a xml based policy doc. From other community owner will enhance the diversification the policy rules.

Editing strategy is easy for anybody that knows the basics of computer

No need for external program to edit and upload policies . Complicates step, may confuse user

Changing the weight of rewards for the action is complex. It would be easy to directly change the content in the selected action tags and weight box

Good luck

Table D-4.1 Suggestions Edit Status Policy.

Scenario: Create a “Transfer Policy” for visitor of a given community

	Over All					Usability					Reliability (crashes etc.)					Overall readability of policy				
	Very poor -2	Poor -1	Satisfactory 0	Good 1	Very Good 2	Very poor -2	Poor -1	Satisfactory 0	Good 1	Very Good 2	Very poor -2	Poor -1	Satisfactory 0	Good 1	Very Good 2	Very poor -2	Poor -1	Satisfactory 0	Good 1	Very Good 2
1					1					1					1					1
2					1					1					1					1
3					1					1				1						1
4				1				1							1					1
5					1					1					1					1
6					1					1					1					1
7				1					1					1					1	
8				1						1				1					1	
9					1				1						1				1	
10				1						1					1				1	
11				1					1						1				1	
12				1					1					1					1	
	0	0	0	6	6	0	0	1	4	7	0	0	0	4	8	0	0	0	6	6
				50.00	50.00	0.00	0.00	8.33	33.33	58.33	0.00	0.00	0.00	33.33	66.67	0.00	0.00	0.00	50.00	50.00

Table D-5.1 Overall Ranking of “Create Transfer Policy "

Scenario: Create a "Transfer Policy" for visitor of a given community

5.1 When a user moves from one community to another, do you think his/her status should be transferred to the second community?

S. No.	Do not agree at all	Not agree	Somewhat agree	Agree	Strongly agree		Don't know
1				1			
2			1				
3			1				
4		1					
5		1					
6				1			
7				1			
8				1			
9				1			
10				1			
11			1				
12				1			
Total	0	2	3	7	0		0
	0.00	16.67	25.00	58.33	0.00		0.00

Table D-5.2 Overall Transfer Policy

Scenario: Create a "Transfer Policy" for visitor of a given community

5.2 Do you think the community owner should be able to set rules for transfer of user from one community to another?

S. No.	Do not agree at all	Not agree	Some what agree	Agree	Strongly agree		Don't know
1				1			
2					1		
3				1			
4					1		
5				1			
6					1		
7				1			
8				1			
9				1			
10					1		
11					1		
12				1			
Total	0	0	0	7	5		0
	0.00	0.00	0.00	58.33	41.67	0.00	0.00

Table D-5.3 Community owner and transfer policy.

Scenario: Edit a “Role Policy” of the community

2.4 Please rank the “Create community screen" of Comtella on a 5-point scale (-2: very poor; +2 very good)

	Over All					Usability					Reliability (crashes etc.)					Policy File Upload					Adding and removing role					Overall readability of policy				
	Very poor -2	Poor -1	Satisfactory 0	Good 1	Very Good 2	Very poor -2	Poor -1	Satisfactory 0	Good 1	Very Good 2	Very poor -2	Poor -1	Satisfactory 0	Good 1	Very Good 2	Very poor -2	Poor -1	Satisfactory 0	Good 1	Very Good 2	Very poor -2	Poor -1	Satisfactory 0	Good 1	Very Good 2	Very poor -2	Poor -1	Satisfactory 0	Good 1	Very Good 2
1					1					1					1					1					1					1
2					1					1					1					1					1					1
3					1					1					1					1					1					1
4					1				1						1			1							1					1
5					1					1					1					1					1					1
6					1					1					1					1					1					1
7					1					1				1						1					1					1
8				1						1				1					1					1						1
9					1					1					1					1					1					1
10					1					1					1					1					1					1
11					1					1					1				1						1					1
12				1					1					1					1					1					1	
	0	0	0	2	10	0	0	0	2	10	0	0	0	3	9	0	0	1	3	8	0	0	0	2	10	0	0	0	1	11
	0.00	0.00	0.00	16.67	83.33	0.00	0.00	0.00	16.67	83.33	0.00	0.00	0.00	25.00	75.00	0.00	0.00	8.33	25.00	66.67	0.00	0.00	0.00	16.67	83.33	0.00	0.00	0.00	8.33	91.67

Table D-6.1 Overall Ranking of “Create Role Policy "

Scenario: Edit a “Role Policy” of the community

6.1 As a community owner, would you like to delegate the task of community administration to other members?

S. No.	Do not at all	No	May be Yes	Yes	Definitely yes
1				1	
2				1	
3				1	
4					1
5					1
6					1
7				1	
8				1	
9				1	
10					1
11				1	
12			1		
Total	0	0	1	7	4
	0.00	0.00	8.33	58.33	33.33

Table D-6.2 Delegation Community Administration.

Scenario: Edit a “Role Policy” of the community

6.2 Do you think delegating tasks will attract more users in your community?

S. No.	Not at all	No	May be Yes	Yes	Definitely yes
1				1	
2					1
3				1	
4			1		
5				1	
6					1
7				1	
8				1	
9				1	
10				1	
11				1	
12			1		
Total	0	0	2	8	2
	0.00	0.00	16.67	66.67	16.67

Table D-6.3 Delegation and Number of Users.

Scenario: Edit a "Role Policy" of the community					
6.3 Do you think delegating tasks will make the community more sustainable?					
S. No.	Not at all	No	May be Yes	Yes	Definitely yes
1				1	
2				1	
3				1	
4				1	
5				1	
6			1	1	
7					
8				1	
9					1
10				1	
11					1
12			1		
Total	0	0	2	8	2
	0.00	0.00	16.67	66.67	16.67

Table D-6.4 Delegation and Sustainability.

Scenario: Edit a "Role Policy" of the community
6.4 The Comtella "Role Policy Screen" offers a default Role Policy. Please, describe in your words what rights the Role Policy assigns.
Owner: can delete link, create com, edit link, edit policy, edit rile : expert can as above Operator can as above….
The rights offers are as follow delete link, create new community, edit an existing link, editing an existing policy and also edit the role of an existing member
The ability to carry out roles deleting, creating a community, editing a link, editing a policy and editing a role
ability to delete offensive material, create new community and branch off, change their own status
The rights allow the member to manage the community based on their roles.
Delete or edit link, create community, edit policy and role
Delete link, create community, edit link, edit role
The owner expert and operator have different level of roles, the owner has all the roles and he can assign his friends to have these roles. The role of operator and expert can be changes otherwise.
The right to delete a link posted by other users, to cr4eate a community, to edit a link posted by other users, to edit an existing policy and to edit existing roles
assign different level of users to do different work
based on his/her status/role a user can perform various tasks as editing policy or roles, links etc delete links and create community.
delete link, create community, edit link, edit policy

Table D-6.5 Remark about Role Policy.

Scenario: Edit a “Role Policy” of the community

6.5 The Comtella “Role Policy Screen” offers a default Role Policy. Do you think this default Role Policy is appropriate for most of the communities?

S. No.	Not at all appropriate	Not appropriate	Somewhat appropriate	Appropriate	Very appropriate
1				1	
2				1	
3				1	
4				1	
5				1	
6			1		
7			1		
8				1	
9			1		
10					1
11			1		
12				1	
Total	0	0	4	7	1
	0.00	0.00	33.33	58.33	8.33

Table D-6.6 Default Role Policy.

Scenario: Edit a “Role Policy” of the community

6.6 The Comtella “Role Policy Screen” allows users to set certain access rights (Delete Link, Create Community and Edit Link/Policy/Role). Do you think these rights are appropriate for the Role Policy?

S. No.	Not at all appropriate	Not appropriate	Somewhat appropriate	Appropriate	Very appropriate
				1	
				1	
				1	
			1		
					1
				1	
				1	
				1	
			1		
					1
			1		
				1	
		0	3	7	2
		0.00	25.00	58.33	16.67

Table D-6.7 Role Policy and Access Rights.

Scenario: Edit a “Role Policy” of the community

6.7 If you have selected “Not at all appropriate” or “Somewhat appropriate” in the previous question, please suggest additional access rights that should be included or removed from Role Policy

Include	Remove
Request to edit policy, request to create new community	Edit policy, Create new community
Create thread, create community	

Table D-6.8 Add or Remove Access Rights.

Scenario: Edit a “Role Policy” of the community

6.8 I found that number of role options (10) for my new community was…

S. No.	Not at all appropriate	Not appropriate	Somewhat appropriate	Appropriate	Very appropriate
1				1	
2				1	
3					1
4			1		
5				1	
6					1
7			1		
8				1	
9				1	
10				1	
11				1	
12				1	
Total	0	0	2	8	2
	0.00	0.00	16.67	66.67	16.67

Table D-6.9 Number of Role Options in Screen.

Scenario: Edit a “Role Policy” of the community

If you have selected "Not at all approropriate" or "No appropriate" in the previous question. Please specify

Too few	To Many
No Response	

Table D-6.10 Remarks.

Scenario: Edit a “Role Policy” of the community

6.10 Would you prefer to edit the Role Policy document in a text editor and upload it with “Role Policy Screen”?

S. No.	Not at all	No	May be Yes	Yes	Definitely yes
1		1			
2			1		
3		1			
4		1			
5			1		
6					1
7				1	
8				1	
9		1			
10					1
11		1			
12	1				
Total	1	5	2	2	2
	8.33%	41.67%	16.67%	16.67%	16.67%

Table D-6.11 Editing Role Policy in Text Editor.

Scenario: Edit a “Role Policy” of the community

6.11 If you found a Role Policy of some other community appropriate for your community, would you like to request the XML based role policy document from that community owner?

S. No.	Not at all	No	May be Yes	Yes	Definitely yes
1				1	
2				1	
3		1		1	
4					1
5				1	
6					1
7				1	
8			1		
9					1
10					1
11				1	
12				1	
Total	0	1	1	8	2
	0.00	8.33	8.33	58.33	16.67

Table D-6.11 Using Role Policy as XML File

Appendix E

This appendix presents the answers to selected questions asked in the questionnaire given in Appendix B and C.

Table E-7.1. Overall communities membership

	Not Motivate at all	Not Motivate	May be Motivate	Will Motivate	Strongly motivate
Will having some special role in a community motivate you to participate more?	16.6%		41.6%	41.6%	
Will having some special role in a community motivate you to invite other people to join the community?		16.6%	50%	25%	8.3%
	Not at all	NO	May be yes	Yes	Definitely yes
Will you be comfortable if someone moves to your community from another community/group and gets a higher status based on his/her status in the previous community?		41.6%	50%	8.3%	

Table E-7.2. Questions about the concept of user status policy and transfer policies

	Do not agree at all	Not agree	Somewhat agree	Agree	Strongly agree
The community owner should be able to set reward rules for his/her community members.				1	11
Would you support the idea of limiting users' actions in the community based on their status (for example limiting who can read, share and comment based on their status)?				9	3
Do you agree with the statement, "Communities may have their own set of reward rules for participation"? (Based on their status)?		1		7	4
If you find a policy of some other community suitable for your community, would you request the XML based policy document from that community owner?		1		10	1
When a user moves from one community to another, do you think his/her status should be transferred to the second community?		2	3	7	
Do you think the community owner should be able to set rules for transfer of user from one community to another?				7	5
If you found a Role Policy of some other community appropriate for your community, would you like to request the XML based role policy document from that community owner?		1	1	8	2

Table E-7.3. Questions about screens of 'Status Policy'

	Not at all appropriate	Not appropriate	Somewhat appropriate	Appropriate	Very appropriate
The Comtella policy editing screen offers a set of actions (Add new share, Give remarks, Rating, Upload file, Post message, Add new community) for reward rules in the Status Policy. Is this set	0	0	1	10	1
The Comtella policy editing screen offers a default status policy with a set of status rules. Do you think this default status policy is appropriate for most communities?	0	0	3	7	2
The Comtella policy editing screen allows users to set certain access rights (share link/file, post message and rating). Do you think these rights are appropriate for the Status Policy?	0	0	2	9	1
I found that number of status options (10) for my new community was...appropriate for the Status Policy?	0	0	1	11	0
	Not at all	NO	May be yes	Yes	Definitely yes
Do you understand the default status policy described in the policy editing screen?	0	0	0	10	2
Would you prefer to edit the policy document in some text editor and upload with Edit policy option?	2	6	1	2	1
If you find a policy of some other community suitable for your community, would you request the XML based policy document from that community owner?	0	1	0	10	1

Table E-7.4. Question about 'Role Policy'

	Not at all appropriate	Not appropriate	Some what appropriate	Appropriate	Very appropriate
The Comtella "Role Policy Screen" offers a default Role Policy. DO you think this default Role Policy is appropriate for most of the communities?			4	7	1
The Comtella "Role Policy Screen" allows users to set certain access rights (Delete Link, Create Community and Edit Link/Policy/Role). Do you think these rights are appropriate for the Role Policy?			3	7	2
I found that number of role options (10) for my new community was…			2	8	2
	Not at all	NO	May be yes	Yes	Definitely yes
As a community owner, would you like to delegate the task of community administration to other members?		5	1	7	4
Do you think delegating tasks will attract more users in your community?			2	8	2
Do you think delegating tasks will make the community more sustainable?			2	8	2
Would you prefer to edit the Role Policy document in a text editor and upload it with "Role Policy Screen"?	1	5	2	2	2
If you found a Role Policy of some other community appropriate for your community, would you like to request the XML based role policy document from that community owner?		1	1	7	4