TAMPEREEN TEKNILLINEN YLIOPISTO

JOONA KANNISTO

Reputation Based Trust Management in a Wiki Environment

Master of Science Thesis

Examiners: Phd Marko Helenius and
Phd Jukka Koskinen
Examiners and topic approved in the
Faculty of Computing and Electrical
Engineering Council meeting on
06.04.2011

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master's Degree Program in Signal Processing and Communications Engineering
KANNISTO, JOONA: Reputation Based Trust Management in a Wiki Environment
Master of Science Thesis, 61 pages.
November 2011
Major: Communications Networks and Protocols
Examiners: Senior researcher Marko Helenius, Lecturer Jukka A. Koskinen
Keywords: trust, reputation, peer review, community, rating

Trust makes our life easier, as we can focus our resources to productive actions instead of verifying the actions of others. Many services in our society would not be possible without trust. We trust products and services that have a good reputation and we recommend good services that we discover to others. Our aim is to find the best selection, taking into account the risk and the benefit. On the Internet the huge number of possible partners and options means a lot of options to choose from. However, online reputation management systems can automatically track our preferences and communicate the necessary recommendations, and applying reputation for trust decisions may be even easier than it is offline.

This thesis seeks to answer what problems in a wiki can be solved with reputation based approaches, and how those mechanisms can and should be implemented. Wikis utilize collaborative writing, and this collaborative approach can create difficulties in generating actor reputation from content. Conducted literature review did not reveal a satisfying solution. Therefore, a reputation management application based on content reviews was made to explore the possible solutions.

Field test for the application was done in a wiki related to teaching material. This test gave insights about user reactions to the application and user expectations for peer-review tools. Reputation related insights did not emerge because the test scenario was relatively short and restricted. However, what emerged was a need to evaluate the comments given as a part of peer-review.

The main result of this thesis is the implemented reputation management application. In addition, the findings show that wiki user reputation and trust decisions can be easily applied to auxiliary functions, such as content reviews and comments. However, larger trust decisions and the evaluation of more complex content needs more work.

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO
Signaalinkäsittelyn ja Tietoliikennetekniikan koulutusohjelma
KANNISTO, JOONA: Mainepohjainen luottamuksenhallinta wikiympäristössä
Diplomityö, 61 sivua.
Marraskuu 2011
Pääaine: Tietoliikennetekniikka
Tarkastajat: Vanhempi tutkija Marko Helenius, Lehtori Jukka A. Koskinen
Keywords: luottamus, maine, vertaisarviointi, yhteisö

Luottamus helpottaa päivittäistä elämäämme, koska voimme keskittyä muiden henkilöiden vahtimisen sijasta tuottavaan toimintaan. Monet yhteiskuntamme palveluista, esimerkiksi valintamyymälät, eivät edes toimisi ilman luottamusta. Luotamme hyvämaineisiin, meille tuttuihin tuotteisiin ja palveluihin, sekä suosittelemme hyväksi havaitsemiamme hyödykkeitä edelleen muille. Tavoitteenamme on useimmiten löytää runsaasta palveluiden tarjonnasta meille sopivin vaihtoehto mahdolliset riskit ja hyödyt huomioiden. Jos haluamme käyttää samaa toimintamallia Internetissä, tarjottujen palveluiden ja mahdollisten kumppanien määrä voi tehdä vaihtoehtojen punnitsemisen todella raskaaksi. Maineen- ja luottamuksenhallintajärjestelmillä mieltymystemme seuranta ja suosittelujen välittäminen muillekin voidaan tehdä automaattiseksi, jolloin näiden mekanismien soveltaminen onnistuu myös verkossa.

Tässä diplomityössä tutkittiin, millaisia ratkaisuja maineenhallintajärjestelmä voi tuoda luottamussuhteiden määrittämiseen käyttäjien välille, sekä sisällön luokitteluun ja suodatukseen opetussisältöä tuottavassa wikiympäristössä. Wikien perusajatuksena on, että sivuston käyttäjät itse luovat sivuston sisällön yhteistyössä. Tämä yhteistoiminnallisuus sisällön tuottamisessa asetti sellaisia haasteita käyttäjien toimien yksilöintiin, joita ei alussa suoritetun kirjallisuuskatsauksen perusteella pystytty täysin ratkaisemaan. Ratkaisuja etsittiin myös käytössä olevista toteutuksista. Näiden ideoiden pohjalta rakennettiin sisällön arviointiin perustuva maineenhallintasovellus.

Maineenhallintasovellusta testattiin kohdejärjestelmässä. Testissä selvitettiin käyttäjien suhtautumista nykyiseen toteutuksen käytettävyyteen ja käyttäjien vertaisarviointiin liittyviä odotuksia ja tarpeita. Pääpaino oli testin lyhyydestä johtuen vertaisarvioinnin toteuttamisessa. Maineeseen liittyviä ominaisuuksia pystyttiinkin testaamaan vain vähän. Suurimmat vertaisarviointiin liittyvät tarpeet olivat tekstimuotoisessa kommentoinnissa ja sen arvostelussa. Tämän havainnon perusteella sovellukseen lisättiin mahdollisuus arvioitavaan ja maineen perusteella suodatettavaan kommentointiin.

Työn näkyvin tulos on avoimena lähdekoodina julkaistu maineenhallintasovellus. Lisäksi työssä havaittiin, että maineen perusteella tehtäviä luottamuspäätöksiä on luontevinta soveltaa wikiympäristöissä sisällön luotettavuusarviointien sijasta oheistoimintojen, kuten tekstimuotoisen palautteen suodatukseen. Suurempia luottamuspäätöksiä tekevän järjestelmän luotettava toteutus vaatisi kuitenkin lisätutkimusta.

PREFACE

This Master's thesis work was carried out in the Department of Communications Engineering, Tampere University of Technology as a part of the Future Internet program of TIVIT (Tieto- ja viestintäteollisuuden tutkimus). The thesis is about online reputation management systems and how can such a system be implemented in a wiki environment. In order to understand reputation systems effect on community, the user reactions to rating other users and content is explored. In addition, the thesis tries to find out what trust decisions can be made with this reputation, and can they be automatic? During this work I have had to rethink my understanding about, such basic sounding concepts as trust and reputation and their meaning to my daily life.

I would like to express my gratitude to my thesis supervisors Marko Helenius and Jukka Koskinen for their excellent feedback during this process. They also contributed to the original ideas leading to this work. In addition, I would like to thank Kristiina Karvonen, who gave me valuable insights to the usability of reputation systems in a TIVIT seminar in Tampere. Special thanks goes to my colleagues at the Department of Communications Engineering for their advice on technical matters and inspiring discussions.

I want to thank my parents for their encouragement and support for the full duration of my studies. However, most of all, I would like to thank my lovely girlfriend Veera for her support and trust in me.

On November 22, 2011, in Tampere, Finland

Joona Kannisto

CONTENTS

TERMS, ACRONYMS AND SYMBOLS

sgn	Sign function
API	Application Programming Interface
CPAN	Comprehensive Perl Archive Network
CSS	Cascading Style Sheet
DCE	Department of Communications Engineering
GPLv2	GNU General Public License version 2
GUI	Graphical User Interface
HMAC	Hash Message Authentication Code
MLDBM	Multi Level Database Manager
JOP	Crowdsourced Teaching Portal
OS	Operating System
P2P	Peer to Peer
RDB	Reputation Database, used to hold user reputations
ReputationPlugin	Reputation management plugin for TWiki
RFC	Request for Comments
SQL	Structured Query Language
TUT	Tampere University of Technology
TVDB	Topic Votes Database, contains votes for TWiki-topics
TWiki	Structured wiki and application platform
TWiki-topic	A page or independent content item in TWiki
TWiki-web	Collection of topics in TWiki
URL	Uniform Resource Locator
UVDB	User Votes Database, contains votes grouped by user

1 INTRODUCTION

We use reputation information for trust decisions as a natural part of our everyday lifes. We ask advice from persons we know to be knowledgeable in areas where we are not experts and use recommendations to find previously unknown resources. A typical case would be to ask recommendations for a good car repair shop if we are currently in a need for a really good one. Our definition of a good car repair shop might be cheap, skilled or preferably both. These mechanisms for evaluating resources are extended naturally beyond physical world to the online world as well. For example, it is common to send links to useful or particularly entertaining websites to friends, and re-visit web services which have previously offered us a good value for our money or time. Online reputation management systems make these actions even easier, as it is possible to objectively compare reputation information and automatically collect and distribute it. While there are many reputation management systems in use in many different areas, the research of online reputation management systems is still relatively new.

The *JOP* (joukoutettu opetusportaali, crowdsourced teaching portal) [1] is a *TWiki*-powered [2] wiki site. The portal is intended to be a community where the users can collaborate in order to produce high quality material. The material is intended to be used primarily as a teaching resource. [1]

The current community is closed in the sense that editing privileges are granted by administration to users who are known beforehand through other connections. However, the intention is to open the editing rights to a wider user base [1]. Online multiuser collaboration environments can have writers with varying intentions and expertize. This can cause purposeful or accidental errors [3]. Therefore, opening the portal to unknown editors can create concerns regarding trustworthiness and quality of the material.

1.1 Online Trust Management by Reputation

When an online collaboration environment grows beyond few people, trust between collaborators is needed as it would be impossible to verify all the actions the other users make. Mayer et al. defines trust as “the willingness of a party to be vulnerable to the actions of another party based on the expectation that the other will perform a particular action important to the trustor, irrespective of the ability to monitor or control that other party." [4] This definition implies that a trust decision carries always a risk, but usually it offers a benefit as well. Benefits of trust in Internet collaboration scenarios are that we do not have to verify every statement individually, and there is a risk that we receive false or useless information.

Reputation is closely related to trust, and these concepts are often mixed up [5]. However, we can choose to trust because of, or despite of entity's reputation [6, p. 214]. Reputation is information about previous actions, and this information can be used to estimate the future behavior and the current capabilities of an entity.

Content and items can have reputations as well [7], and these reputations can be used to make estimates about their usefulness. Reputation information is used for many decisions and it is evaluated more strictly when important, and possibly costly, decisions have to be made [7, p.129]. Reputation can be communicated as well and Whenever we do not have direct observations about all the possible choices, we estimate entity's reputation from recommendations gathered from our trusted peers.

Online reputation management systems can be used to manage trust in situations where some trust between collaborators is needed but traditional methods for evaluating the trustworthiness of other users are impractical or undesired. Sometimes the sheer number of participants in an online community makes it impossible for a human to keep track of their every relationship. Furthermore, in most online communities there is no easy or socially acceptable way to request recommendations about an unknown peer from previously known peers [8, p. 11]. Online reputation management systems have been created to address these problems by automatically tracking reputation and managing the recommendations needed to evaluate unknown peers.

1.2 Requirements and Limitations

This thesis was done to research and improve reputation management features in the scope of JOP. The work was based on the author's earlier bachelors thesis work on the same subject [9]. The goal was to improve the existing work and to test it in the target environment. Main scope of improvements was in the usability and code cleanliness, although additional mechanisms related to reputation management were implemented as well. Besides being extended and refined, the previous concept's fitness to the task was re-evaluated. Theoretical considerations, and results from a pilot test organized on the target platform were used for this.

When implementation related improvements were discussed with the thesis instructor Marko Helenius, improving the existing implementation's usability was the most important issue. The original interface [9] was very plain, and it contained few usability related bugs, such as changing order of the voting buttons and random order of elements in the returned listings. In addition, the auxiliary interfaces were generally unfinished, they contained terms which were confusing and very specific to the application. Usability and appearance are very important for product acceptance, since for the end user they are usually the first and in some cases the only things that matter. As the final product is a security related addon anything related to its usability has to be evaluated with that in mind. For example, user's primary focus is not usually in using this tool, but it should be there when needed.

The limitations and strengths of the platform, and especially its plugin API, largely defined what could be implemented without going to unnecessary extremes. The original implementation was intended to test implementing a reputation management system in TWiki environment. However, all the planned use cases for the product were not strictly related to reputation management, as there was need for peer review tools as well.

The implemented system, named ReputationPlugin, was a general reputation management system, with social network filtering. The intention was to provide both a useful content rating system for the casual visitors and users, as well as an engaging reputation management and feedback system for the regular contributors. The plugin source code was published under an open source license, namely GPL version two (GPLv2) [10], on TWiki.org [2], which means that anyone can make modifications to it and distribute the resulting work under the same license.

1.3 Methods

This thesis consists of a theoretical part, and a constructive research part including the implementation of a reputation management plugin and a short field test of it. The theoretical part is largely a literacy review including a survey on existing implementations. The constructive research part describes the implementation details and decisions as well as the reasons behind them. The field study was organized in the target environment, as a partially controlled study on the functional elements. To map the theoretical background to the implementational side, the key findings from the constructive research part are analyzed in detail and compared to existing solutions.

1.4 Structure of the Thesis

The second chapter describes the study field: wiki environments, reputation management systems and usability goals for security management software in detail. In the third chapter known solutions and existing implementations for similar problems are presented and their ideas are evaluated in the context of the target environment. The fourth chapter presents the design choices and the reasoning behind them. Technical description of the built system is also in the fourth chapter. The applicability to the problem and ideas for future development are discussed in the fifth section. Conclusions are provided in the final section.

2 BACKGROUND

2.1 Trust and Reputation

Trust is a human tool for hiding the complexities of risks and benefits present in actions involving another entity to a simple abstraction. The existence of risk and benefit are necessary for trust, as trust is unnecessary in fully controllable situations, and if there is no benefit there is no incentive for the trust to develop. Trust decisions involve the trustee, who is the entity trusted with the asset and the trustor who makes the trust decision and is in control of the asset. While trust is a human tool, computer systems can be made to mimic human trust.

Our society needs trust to function, as without trust only a very limited set of possible transactions can occur. Trust can also make some possible transactions more efficient. [8, pp. 1 – 2] For example, we can walk to a convenience store and the store manager trusts us to pay for the products we take as we leave the store, and does not control the situation by watching every single action the customers take. This saves the store manager's time, and makes it possible for the store to serve more customers. The benefit from reduced work far outweighs the costs coming from occasional shoplifting.

Trust has a few general properties. Trust is an unidirectional property, but mutual trust exists in the sense that sometimes trust creates trust. While certain properties of trust are transitive trust itself is not [8]. In addition, trust is never absolute, but instead it always has a context and an amount [6]. For example, I might trust my neighbor with keys to my apartment while I am on a trip, but I would not trust him to fix my car. In a similar vein there might exist a mechanic to whom I would trust my car, but not necessarily the keys to my apartment. Trust decision involves both perceived capability and reliability, and the exact need for them comes from the risks associated with the asset and the trustee [6].

Reputation is a communicated value of an entity's cabability and reliability [6]. Reputation and trust are related, as reputation is a commonly used metric in evaluating trustworthiness. Use of reputation in trust decisions is based on the idea that past behavior is indicative of the future behavior. Reputation can be used to estimate both capability and reliability. Reputation can be communicated as recommendations. Like trust, reputation is tied to a context and, reputation in one area is not directly transferable to other areas.

2.2 Trust and Reputation Management

Trust management systems which use reputation are in the soft security area of information security taxonomy. Hard security uses strong methods to grant absolute permissions, but it fails completely if it is bypassed or applied incorrectly. In contrast, soft security allows actions as long as they conform to good behavior. [11] These two approaches are not directly competing with each other, and usually both methods can be combined for the best possible outcome. For example, a social firewall application could utilize the decisions made by reputable expert users to provide the same level of security to users who are incapable of making the decisions by themselves [12].

Trust management through reputation has its roots in the electronic commerce. Mutual trust between trading partners is needed as the online environment makes it impossible to inspect the goods and ensure their delivery [13]. These two uncontrollable risks are focused on buyers, but sellers face some risks as well. The risks that sellers face are mostly related to scammers imposing as legitimate buyers and trying to either get the item for free or have the seller send them money by using fraudulent payments. The most common risk to a seller, although not as severe as the former risks, is that the buyer will back out from the deal wasting the sellers time, and possibly also the listing fee paid to the online auction site for the right to sell the item on the site.

Reputation can be extended from entities with free will to things too [7, p. 4]. This type of reputation is not used to estimate future behavior of the object, but the result of our interaction with it. This extension is necessary when we move from the domain of e-commerce to the domain of content production. The resource which is protected in the content domain is the user's time and the site's resources. With reputation it is possible to filter and highlight both interesting users and content as well as flag offending content and malicious users.

Identity management in online world is a problem area where reputation management systems can offer a solution to a controversial problem. A common example of this is online auctions: on the other hand there is the desire to remain pseudonymous or even anonymous for privacy reasons, and on the other there is a desire to prevent fraud. Using real life identities, authenticated by a a central authority, instead of user generated pseudonyms would mean that one person could not have multiple accounts and that misbehaving users could be punished very easily.

However, such a system would be very costly and include privacy concerns. For example, sales of certain sensitive items would see a decline in online auctions. Furthermore, in many cases using the real life identities does not have large benefits over pseudonyms, as the reputation of a person is not likely to be known to another person living on the other side of the world. In this sense, reputation management systems are an excellent way of building specialized communities with trusted pseudonyms. If building a good reputation is necessary for a pseudonym to be trusted, the reputation management system can impose a cost for creating a new pseudonyms without exposing users true identity to the target site or the other users [14] [15]. The only requirement is that building the reputation has a cost. This cost can be either monetary or a time and effort based one.

2.3 Wiki Sites

Wikis are web sites which consist of user editable content. The pages in a wiki can be edited through a web browser with a relatively simple syntax. This syntax is not consistent among different wiki platforms, but it usually gives users fast access to common text styles and interwiki link creation. Because the editing is simple and fast, wikis have become a preferred method for many projects requiring collaborative writing. Wikis differ from more general content management systems in that they are usually intended to be edited by their users.

At the time of the writing[1], the most famous wiki site is Wikipedia, which is an encyclopedia that anyone can edit. Wikipedia does not restrict the edit rights to content [3], but instead tries to achieve high quality and reliability through openness [16]. Restrictions are placed only when repetitive vandalism is detected [16]. The intention behind the openness is that when everyone is allowed to edit the content, the errors can be corrected fast [16].

In contrast to openness of Wikipedia, many wiki sites are not open to the public. Restrictions on the editing rights are more common, but read access can be limited as well. For the purposes of access control, typically a few different forms of identity management are used on these restricted wiki sites. Some wiki sites allow outsiders through invitations generated from the wiki itself, and some rely on separate channels, such as personal email to the administrator, to gain access. Typical example of a restricted wiki is a project wiki deployed to a company's intranet, which is not visible to the outside world at all, and which uses company's established identification and authorization mechanisms.

Besides the user access controls and identity management properties, wiki sites

[1]April 2011

have differences in the content as well. Some wikis serve as scratchpads for new ideas, while some are intended to be used as the means of final publication. This distinction affects the requirements for the quality of the information as well as the clarity of the presentation.

2.4 JOP, Crowsourced Teaching Portal

2.4.1 JOP Description

The purpose of JOP is to be a community driven central database for teaching material in the field of information security. The material is intended to be gathered from multiple sources, and structured to be reusable by different users and in different contexts. This is accomplished by splitting the material into smaller elements, each containing a self-standing part of a larger topic in some area. To keep these fractions of a topic organized they are classified. Classifications are made according to the complexity, category and completeness of the material.[1]

The JOP in its current form is a partially restricted wiki site. User accounts can be registered by anyone, but these accounts do not have rights to access the restricted parts of the wiki. The identity management needed to grant additional access rights is handled mainly by using existing course registrations and manually mapping the registered user accounts to groups with course specific access rights. In addition, users can request access rights from the JOP's administrator group [17].

The requirements for trust in JOP are related to content and the authors. It would be desirable to give trustworthy authors edit rights to content and find the best content they produce. As the content in JOP is primarily teaching material, it has two main requirements: it has to be reasonably up to date and reliable. These two goals for the quality of material can be conflicting, as the freshness of material can greatly benefit from openness while the reliability and overall quality may suffer.

User base of JOP was intended to consist of different expertize levels and areas [1]. Two main expertize level groups were teachers and students, although other groups were intended to be included as well. Both teachers and students could participate in producing teaching material, and instead of unidirectional information flow from teachers to students the goal is to build a collaborative environment with possibilities of feedback. In addition, the community membership is intended to extend beyond graduation and therefore even economic incentives can affect the content of the topics.

2.4.2 JOP Technical Platform

TWiki [2] is a wiki application written in Perl. TWiki, apart from other wiki software alternatives, was chosen as the platform for JOP because of its advanced access control properties, open source development model and lack of severe security problems [18, p. 13]. TWiki's access rights can be controlled either at the web or topic level through group and user specific allow and deny lists. These access control settings are adequate for most use cases. Besides the advanced user rights management TWiki's main features are hierachical structure and good support for extensions. Hierachical structure is formed by separate workspaces (*TWiki webs*) consisting of individual pages called topics (*TWiki topics*).

Any additional functionality for JOP had to be provided with either custom made or existing TWiki Applications [19]. TWiki Applications consist of integrated TWiki topics and TWiki extensions [19]. TWiki extensions provide additional functionality without affecting the TWiki core. They are commonly referred as TWiki plugins and consist of a Perl module and a documentation topic [20]. The plugins are written in Perl [21] and can use all the Perl's features as well as external Perl modules. For interaction with TWiki the plugins are instructed to use the TWiki plugin *API*, which provides a stable interface to TWiki's functions.

2.4.3 Reputation and Trust Management Needs in JOP

The need to manage reputation comes from the desire to have a large community. Reputation can function as an incentive, and a reputation management system can be used to provide additional value in the form of content and user filtering. These filtering cababilities are relevant in terms of trust management, since the asset which is protected is clearly the user's time as well as the site's resources.

Exercise works are an example of a resource which is not utilized to its full potential. Students write good quality documents as a part of their normal course work, but these are hidden in archives after they have been reviewed. However, exercise works in fast changing subjects have important value as teaching material. But utilizing this resource needs a way to sort the contributions by quality and feedback functionality for enhancements.

In addition, pseudonymous identity management is possible through reputation. Because much of the material in JOP is public, some contributors will have a desire to stay pseudonymous. This is currently not recommended, and users are encouraged to use their real names as it helps the administration. The challenge is to have a safe place for users to build the necessary reputation for a pseudonym to be accepted as a portal contributor. An additional desired feature would be a mechanism to transfer the reputations from the real world into pseudonyms used in the portal and back.

2.4.4 Community Aspects

Currently most of the topics in JOP are written as compulsory course assignments. User engagement with the portal is required to have more voluntary contributions. One of the easiest ways to create engagement is to offer a feedback mechanism. Users receiving feedback for their initial contributions are likely to contribute more in the future. [22]

When building a reputation management system for community purposes it is important to look at the incentives for user participation. Identifying the motives for contributions helps providing incentive to operate in the desired way. Contributors can have different motivating factors, and the actions done to promote some of these incentives can sometimes demote other incentives. Farmer lists three incentive classes for user participation, these are: altruistic, egocentric and commercial incentives [7, pp. 111 – 120].

Altruistic incentives are the most interesting as they can result in deep engagement in the community and high quality contributions. Ariely [23, pp. 75 – 102] states that people with altruistic motives can work as hard as people with other incentives, and harder than people with weak commercial incentives. However, altruistic incentives are in general harder to maneuver than the other motives. If altruistic incentives are the main motivating factor, there is not necessarily need for any additional rewards. On the contrary, people with altruistic motives can find rewards and attention distressing. The social need to return favors, called pay-it-forward incentive, can be promoted quite easily by offering a valuable resource. This incentive is more powerful if there is a perception of a peer relationship [23, pp. 75 – 102].

Egocentric incentives revolve around recognition for contributions, and they can be amplified by things such as scoreboards and merit badges. These must be used sparingly and only for sufficiently complex actions as they can create undesired behavior, or in some cases unwanted level of desired behavior. There are egocentric incentives which are related to self-fulfillment, and promoting them is not as likely to lead to negative consequences. [7, pp. 111 – 120]

Promoting egocentric recognition incentives can create a feeling of ownership over a certain topic. Ownership of a topic is both a bad and a good thing, as it is good that someone takes the responsibility for the quality of a topic, but bad when someone imposes their own strict viewpoint over a questionable matter. [3] Despite their risks, ownership incentives would probably promote long-term contributions more than any other incentive.

If we think of course credits as currency, then it is clear that there are commercial motives, as some of the users will think of completing assignments as revenue. If these rewards are offered to the community members automatically it is likely that some of them will turn to shorthanded measures. In addition, if these rewards are given for community participation, it is likely that market norms will replace social norms and voluntary contributions will vanish [23, pp. 75 – 102]. However, as the same platform is supposed to work as a community platform and an exercise work platform this problem cannot be completely avoided.

Essentially the desired incentives are based on social values. The personal fulfillment from completing a task and recognition from peers can be seen as the most easily promoted healthy incentives. However, the promoted incentives are not the same for the whole JOP, as there are separate sections and distinct user bases. Moreover, it may not be desirable to have everyone as content creators. Indeed, most users are there to read existing information and they can have different role in the reputation management system and can contribute by giving feedback. However, in advanced courses, these roles are not necessarily separate and peer review capabilities are needed as well.

2.5 The Security of Reputation Management

Reputation management systems divide into centralized and decentralized systems. The difference between these systems is that centralized systems have a trusted authority that collects all the reputation data, while decentralized systems store reputation data in individual nodes. Decentralized reputation systems are suited to environments which are decentralized by their nature, such as ad hoc networks and distributed P2P networks [24]. Centralized models are much simpler to implement and should be preferred if a trusted authority exists. Using central trusted authority for storing the reputation data and providing authentication services eliminates many threats which have to be considered in decentralized systems, such as repudiation, false statements and reputation theft.

In addition to the way reputation information is collected and stored, there are also differences on how it is evaluated. There are two main approaches to evaluating reputation. Reputation can be considered as an individual reputation usable by a certain observer only, or as a more global measure evaluated over predefined set of observers. The term global reputation is used even though reputation is always meaningful only in a specific context [7]. Direct individual reputation, where reputation is only based on trustor's own past experiences can be considered to offer highly reliable and relevant reputation information, but it offers the least benefits as there is no information about new possible partners. However, it is possible to use recommendations from reputable sources to reduce the need for direct observations, without losing the credibility of the observations completely. Global reputation models have high performance as they do not have to be evaluated for each user separately, but they lose some of the context and reliability. In a sense, the global mode can be considered to be the individual mode without the possibility of limiting the recommendations on a per user basis.

ENISA's security analysis for reputation management systems [24], lists 15 threats affecting common reputation management system implementations[24, p.3]. Most of the threats affect only decentralized systems, but a few of them have to be considered in the scope of the JOP as well:

- Whitewashing attack
- Sybil attack
- Denial of reputation
- Extortion
- False statements
- Privacy threats
- Group-think

The use of unrestricted recommenders makes global reputation models vulnerable to sybil and whitewashing attacks, especially when the cost of additional identities is low. In whitewashing attacks the attacker abandons an unreputable identity and gets a new one with a clean reputation. This attack can be countered by assigning low initial reputation for new identities or by keeping the reputation of an entity a secret so that an attacker cannot determine whether their current account has positive or negative reputation. In sybil attacks the attacker creates multiple false accounts and uses them to boost the reputation of a primary account. Countering this attack requires a cost for creating pseudonyms, or pseudonyms which are not trusted by default.

Threats such as denial of reputation, extortion and false statements can be considered to be semantic attacks, and they cannot be completely eliminated if the users can express their opinion freely. The attacker plays according to the reputation management system's technical rules but tries to exploit semantic weaknesses in the systems design. These attacks are sometimes called *gaming* attacks. The feasability of these attacks can depend on the target site's demographics and from the rewards and punishments that the reputation management system is connected to. False statements, which can be either positive or negative and denial of reputation can happen if there is incentive in the system to not give out reputation statements or

In extortion the target is given bad reputation by the attacker unless the target complies with attackers demands. This can be a threat in systems where there is substantial punishment from bad reputation or an individual or a small group has large influence on a global reputation system.

Groupthink, which means that opinions are strongly influenced by the assumed group opinion and alternative viewpoints are rejected strongly, targets the newcomers, who might not share the community's opinion. The phenomenon is understandable as it can be seen as the community's way to protect itself from potentially harmful influences. Reputation management system design has only limited ability to reduce groupthink. One way to avoid the situation where group follows the opinion of few reputable persons is making the personal reputation less visible.

Privacy threats are a concern if there is an expectation or perception of privacy. However, if the information is assumed to be public the decisions are more likely to be influenced by groupthink and the barrier of entry will rise. Recommender systems can combat privacy issues only by imposing limits on outgoing recommendations. However, these limits can create bootstrapping problems.

2.6 Usability Requirements

Trust and reputation management systems are security systems. Usability is an essential part of security systems, as systems which are not easy to use cause users to make mistakes. In addition, security solutions are disabled by the users if their value seems to be less than the effort required to use them. [25] Despite this inherent need for usability, security applications are infamous for their unusability [26]. Part of this problem comes from the fact that security applications tend to prevent something bad from happening rather than make something good to happen. This form of negative requirement presents challenges for verifying the correct operation of the application as well as making the application desirable to use [25]. For these reasons, providing additional value to the user in a safe way was set as the main goal.

In addition, users do not usually use security applications actively. In many cases user's primary goal is to do something else, and the security application just gets into their way. [27, p. 22],[26] [28, p. 682] When this is the case, user interaction should be avoided in order to prevent making a wrong choice due habituation or stress condition. [27, p. 583],[26, p. 23-24] For example, user input should be gathered through voluntary actions instead of blocking the user's view with something they are not necessarily interested in. In many cases the needed information can be gathered from actions themselves, without explicit interaction. However, there is a strong argument for making security systems visible, in order to keep the users informed about their inner workings and prepared to manage their own security, should the automated approaches fail. [29] As there are benefits from removing the human component [30], the implications to effective security must be evaluated case by case basis.

All of the usability problems present in security related applications are not caused by their inherent complexity, their nature as secondary activities or the existence of negative requirements. Instead, many of them are those present in all applications, which are not developed with usability as one of the primary goals. These problems can, however, affect security applications more than others as correctly applying security mechanisms is often critical to their effectiveness. Some of these problems have been overlooked in the past, as the target audience for security applications has traditionally been computer savvy users and military applications and they have leveraged their user base only recently. In addition, development efforts are not always geared towards usability, as measuring usability and the effective security it creates is not easy, and it is common to advertise and concentrate on more easily measured things such as encryption strength. [26]

3 RELATED WORK

The reputation management systems with the largest user base are currently those developed on a trial and error basis and used for a single site only [31], [32]. This is partly because reputation management systems are not easy to "bolt on" and the target system may require modifications [15]. Therefore, developing a reputation management system to an already existing target system can be more cost effective than modifying the target system to fit the needs of an existing reputation management solution. In addition, research in reputation management systems is fairly new, and it still tends to encounter issues which have been resolved by trial and error in practical implementations.

3.1 Reputation in E-commerce

Perhaps the first widely accepted and used commercial reputation management system was the reputation management system on the online auction site *eBay*. In online auctions, individual sellers can list items they are willing to sell and potential buyers can bid on them. It is customary for the buyer to send money before receiving the item in question, and to make the transaction the buyer has to trust the seller. To help buyers find reputable sellers a reputation management system was created.

In the eBay reputation management system the users of the site can give feedback to each other after a completed transaction. The feedback is given as a grade (negative, neutral or positive) and it is possible to include textual feedback as well. The gathered feedback is shown as the sum of negative and positive feedback and the amount of negative feedback is shown in parentheses if it exists. This system is relatively simple, but it has shown to be beneficial to the users. While the reputation management features that this system offers might be considered inflexible and to a certain extent unreliable [33], the main benefits it offers are identity stabilization and a sense of community [27, p.77]. The sense of community means a smaller perceived risk, which helps to create the needed trust for trading. Because eBay gets its revenues from comissions it has an incentive to make a system that promotes trust.

The eBay reputation management system uses a global mode of reputation, which has performance benefits as well as lesser requirements for inter-connectivity between the users. The downside of this model is that it does not define any scope for the reputation and so it can sometimes lack necessary context. For example, an eBay user could gain a good reputation score by selling small value items, while simultaneously defecting on transactions concerning more expensive items. Similarly fake transactions through fake accounts can also be used to boost the reputation value. However, in eBay gaming with fake accounts takes time and costs money, as eBay charges comission for every transaction, and transactions are required for feedback. Indeed, the barrier has been high enough, as scammers usually do not utilize complex schemes, but instead aim for easier targets, and fraud which does not attack or game the reputation management system is still more common.

There has been incentive to develop more accurate and not as easily exploitable reputation management systems for e-commerce. Two reputation management systems represented by Zacharia et al. [34] *Histos* and *Sporas* can operate on the same data concurrently, but they build a different view of it. Sporas is based on the more traditional global model of reputation, while Histos is based on the web of trust model. In the web of trust model recommendations are accepted only from trusted sources and weighted according the reputation of the recommending agent. Sporas is intended to be used in the bootstrapping phase and as a fallback mechanism, if Histos does not have sufficient amount of data for a useful evaluation. These reputation management systems were developed for online market place Kasbah, which utilizes automated agents in the selling and buying process. These agents negotiate the price of items without active user involvement beyond initial rules for the bargaining process [35]. Kasbah has not been commercialized and this reputation management system does not have any active use.

3.2 Trust and Reputation for Content

This thesis focuses on the reputation of the content as well as the reputation of the content producers, who are in this case the users of the site. Many existing reputation management systems either use strong separation between these two concepts, or keep them tied together. Are the users seen by the community through the content that they produce, the actions they make and the reviews they give, or can they communicate in other ways, such as home pages describing information about the user and separate discussions between the users? The additional information can affect their reputation among their peers, but it might not affect the reputation of their content. Therefore, a distinction has to be made whether the reputation management system should reflect the human view of reputation, which includes variety of information sources, or to construct a perhaps more objective but narrow view through the use of more limited data.

Slashdot [32], a popular news and discussion site aimed at information technology professionals, uses a content rating scheme to filter out unrelevant information from discussions. Individual posts can be rated by voting them up or down with a one word category rating explaining the decision. Categories for down votes are *offtopic*, *flamebait*, *troll*, *redundant* and *overrated*, while *insightful*, *interesting*, *informative*, *funny* and *underrated* are used for positive modifiers.

In this system a group of trusted moderators is used to rate the information. The trusted moderators are partly chosen from ordinary users with good reputation, and a few of them are paid moderators working for the site. Good reputation, referred as *positive karma*, can be gathered by making relevant comments and participating in the reputation management system itself. Slashdot tries to overcome the problems related to sock puppet accounts by making the peer review capabilities of ordinary users limited, and by using a system called meta-moderation, where the correctness of moderations are evaluated. Meta-moderations themselves are evaluated statistically to prevent abuse. The main limits for becoming a trusted moderator are based primarily on the age of the pseudonym and its reputation. Visiting activity on the site has an impact as well, as those with only infrequent site views and those who use the site obsessively are filtered out. [32]

In addition to the global reputation management system, users on Slashdot can form friend relationships to other users. Adding friends does not affect the global reputation, but instead modifies only the affected users' comment views by giving an option to use an extra comment visibility modifier for friends. The friend relationships are unidirectional, but the user who has added someone as a friend appear to the target user as a fan. It is possible to use an extra comment modifier for friends of friends as well. The opposite for friends are foes, and they can be given comment visibility modifiers too. All the friend and foe information is public.

Because Slashdot is targeted at technology professionals, the reputation system on the site has been a desired target for technologically capable malicious users. Numerous attempts have been made to game the system, and because of that the system has been developed to be as robust as possible. This has limited some of the possible functionality. In addition, Slashdot is a high traffic site, and performance aspects have been a great concern as well.

Adler T.B. et al. [36] have developed a system for assigning trust to Wikipedia content. This system compares the lifetime of users' edits, the period of time that a particular content stays unedited on the site, and assigns a value of trust to individual text excerpts based on it. Authors gather reputation from the amount of trustworthy text they produce and this reputation is used in determining the trustworthiness of new text they produce. Author's reputation has an effect on the trustworthiness of other authors' text excerpts on pages that they have edited, by making the text they left unchanged more trusted. The main benefit of this trust and reputation management system is that it does not need any active user involvement to work.

The Wikipedia trust tool includes a user interface for viewing the assigned trust ratings. The user interface presents an alternative view mode to Wikipedia articles where different parts of the text are marked with different trust values. The values are coloured with a common colourscheme, from green meaning trusted to red meaning untrusted.

Like the Wikipedia project itself [3], this system does not take into account different expertise levels or areas, but instead considers all texts to be equal. Author reputation has no context. In addition, much like the Wikipedia project itself this system needs a high volume site to be attack resistant, as the content driven algorithm relies on the crowd for determining the trusted text and the effort needed to poison the system depends on the number of well-behaving volunteers and administrators. Wikipedia has a massive number of volunteers as well as additional controls to stop automated attacks.

Wikipedia project started recently [1] a feedback system pilot, Article Feedback Tool (AFT), for evaluating the quality of Wikipedia articles. AFT's main intention help article editors improve their articles. In addition to content ratings, the AFT pilot tries to find out whether feedback possibilities lower the barrier of entry for new users. AFT was also intended to be used for content flagging. [37]

3.3 Reputation Management for TWiki Platform

One of the goals in the current work was to build a community, and it was considered that this goal was best met with a peer review scheme instead of pure credibility analysis based on content analysis, interlink calculation and other indirect reputation inputs. The idea behind an interactive component was to lure the content consuming users to interact with the portal, by lowering the barrier of entry to a single click.

As the goal was to implement reputation management features specifically to the JOP, the reputation management system needed to function as a TWiki plugin. A few different rating related plugins existed on the TWiki platform, most notably PeerPlugin[38], VotePlugin[39] and TagMePlugin[40]. Because of author's inexperience with Perl the implementation was decided to be built using existing code as much as possible. All of these plugins were published under GPLv2, which meant that they could be modified freely. When choosing the most promising starting

[1] July 2011

point, the considered factors were code simplicity, storage model and the amount of work needed to extend the plugin with needed functions.

PeerPlugin had the most fitting description, but its development had stalled and it required an external database engine, which was considered to be an unwanted dependency. As there was no guarantee that the PeerPlugin would install correctly to recent TWiki versions, and it required complex steps to set up, it was not considered further. VotePlugin [39] had a great interface, good feature set and an active development community. However, this plugin only provided interface and storage functions for individual polls, and as it did not have suitable storage mechanisms for aggregating data from multiple polls together it was abandoned.

The most promising feature set was offered by TagMePlugin, which was primarily made for content classification, but at the same time it could be considered to be a content popularity and relevance measurement tool. Because this plugin offered a simple storage model and consisted mostly of simple and straightforward code, it was used as the base in the first phase [9]. In this work this implementation was improved and extended.

4 DESCRIPTION

4.1 Reputation Management Framework

The most important part of the reputation management system is the reputation framework [7, p.297]. This part is accountable for the reputation calculations from the data gathered, and the output that the system provides depends on it. The used inputs and their weight on the reputation depend on the reputation framework as well. For example, a system using page views and votes as reputation inputs for content reputation should weight the individual reputation inputs differently, while only an aggregate score might be displayed.

The traditional area of interest for reputation management systems has been electronic commerce, but the ideas can be extended to be used in other environments as well. The reason why electronic commerce has been the main research area in reputation management is that the risks can be easily defined, as they are of monetary value. In addition, transactions in electronic commerce usually have only two participants, which makes accountability simple. In contrast, collaboration environments do not have as clear transactions, and implementing reputation management concepts from electronic commerce to collaboration environments is not trivial since deception and successful co-operation are not always clear. Pure vandalism is easy to detect but the more subtle deceptions like providing false information about a competitor on purpose or false statements as a result of misunderstanding are not so easy to detect and verify.

Essentially, most wiki environments provide enough data for a reputation management system. Revision history and user accounts are practically required for a wiki to function, and as a consequence the user actions can be tracked down very precisely. However, there are challenges in using automated approaches in analyzing this information.

4.1.1 Accountability

The collaborative nature of wikis makes evaluating the quality of individual contributions very hard. Firstly, the contributions cannot be evaluated without the supporting context, and secondly evaluating the whole text does not take into account the amount of relative contributions to the text. While evaluating the amount of textual changes in the revision history can be helpful in determining the major contributors, it is a quantitative approach and as such it fails in situations where the text is rewritten, but the original ideas still remain. Apart from vandalism, every author still deserves credit for a good text. Bad content can be a product of multiple authors as well, and some authors may have done only minor corrections without motivation to change the whole underlying text.

Because of such difficulties in the accountability of the author reputation for wiki content, neither the rewards or punishments for single actions were made substantial. Instead they were intended to be cumulative over multiple topics and all the authors of a topic get reputation whenever a topic is voted. The downside was that it was still relatively simple to gain personal reputation by making minor modifications to multiple popular topics. Indeed, in the current implementation, the author gets credit even if the added text is removed altogether. A content driven approach such as the Wikipedia trust tool by Adler T.B. et al. [36] would have been resistant to this, as the determining property is the quantity of the text and the time it stays on the main version of the page.

4.1.2 Changing Content

Wikis are easy to update, and because of this they are in constant change. This, on the one hand creates a need for trust management tools and on the other hand makes implementing them more difficult. Text that has been previously validated as trusted may change, and that can invalidate the previous evaluation. However, if the system were to completely forget the previous evaluations and treat the newer revision of the text as a completely new text, the value in previous evaluations would be lost. If the topics are evaluated instead of individual contributions, there has to be a balance between amount of data and its accuracy.

One solution is to use the reputation of the last author to filter the inherited scores, as an author with a good reputation can be considered to improve the text and to fix factual errors. One cannot make much assumptions about authors with bad reputation as they can be either malicious or incompetent, but it is likely that they would carry at least some of the good properties of the text to the newer revision, while retaining a higher portion of errors and possibly even adding some. Weighting the votes slightly according to the reputation of the last author can be applied iteratively from the first version to the last.

While the previous consideration could be used, a perhaps more simple and reliable property for the user is whether the subsequent authors are on have a similar reputation. If two authors are approximately similar from the user's viewpoint, the ratings given to the first should apply to the other as well. In the case of two dissimilar authors, we can conclude that we should not transfer the ratings. Values in between can be dampened by the distance in reputation between the authors. Equation 4.1 shows a formula that uses the linear distance in trust value between authors to calculate the inherited score for revision k. The $Score_j$ is the evaluations given to revision j and T_j is the reputation based trustworthiness $T_j = -1..1$ for the author of the revision j. This evaluation can be used independently of evaluation of the credibility of the votes themselves.

$$\text{Total}_k = \text{Score}_k + \sum_{j=1}^{k}(1 - abs(T_j - T_{j-1})/2) \times \text{Score}_{j-1} \tag{4.1}$$

However, the scheme in equation 4.1 was not implemented because of performance concerns, as the TWiki version history would have to be searched for every page view. Collecting the author information with the votes would be necessary for implementing this feature efficiently.

The simplest solution is to simply "forget" the oldest ratings as the revisions change. This can be done either by taking into account only the ratings for a couple of last revisions or by applying a weighting scheme where older votes have smaller influence. These are both suboptimal solutions as the change in revision number does not fully reflect the changes in the text. However, as revision numbers were easy to collect, a simple revision number based weighting scheme could be implemented in the plugin without any extra work. In this scheme the old ratings for a topic were divided by the difference of their revision number and the current revision number. This weighting scheme has more dramatic effects in the beginning, where revision numbers are still small. This is desirable in some cases as the most substantial changes to the topic tend to happen between the earliest revisions. Another way to achieve the same results, but on the global scale, would have been to add a running counter value to every vote and to simultaneously update a central counter value which could then be used to weight the votes.

Other aging functions, such as time or read-access-based aging could have been used, but as the original version of the plugin did not store the time or read-access statistics, they were not implemented. In addition, many other options were discarded due to lack of time. For example, the change in the length of the topic's text would have been a simple and probably very efficient aging measure.

4.1.3 The Implemented Reputation Model

Ratings can be calculated from the votes in various ways. The simplest way is to add the votes together taking into account their numerical value and present the result to the user as a number, this mode represents the global reputation model where all the other users are equal. The score can also be calculated using reputation information, where votes from trusted users are counted with a weight relative to their trustworthiness. Reputation information can be used with or without recommendations, and the threshold for trusted voters and trusted recommenders can be changed by the users. (Figure 4.1).

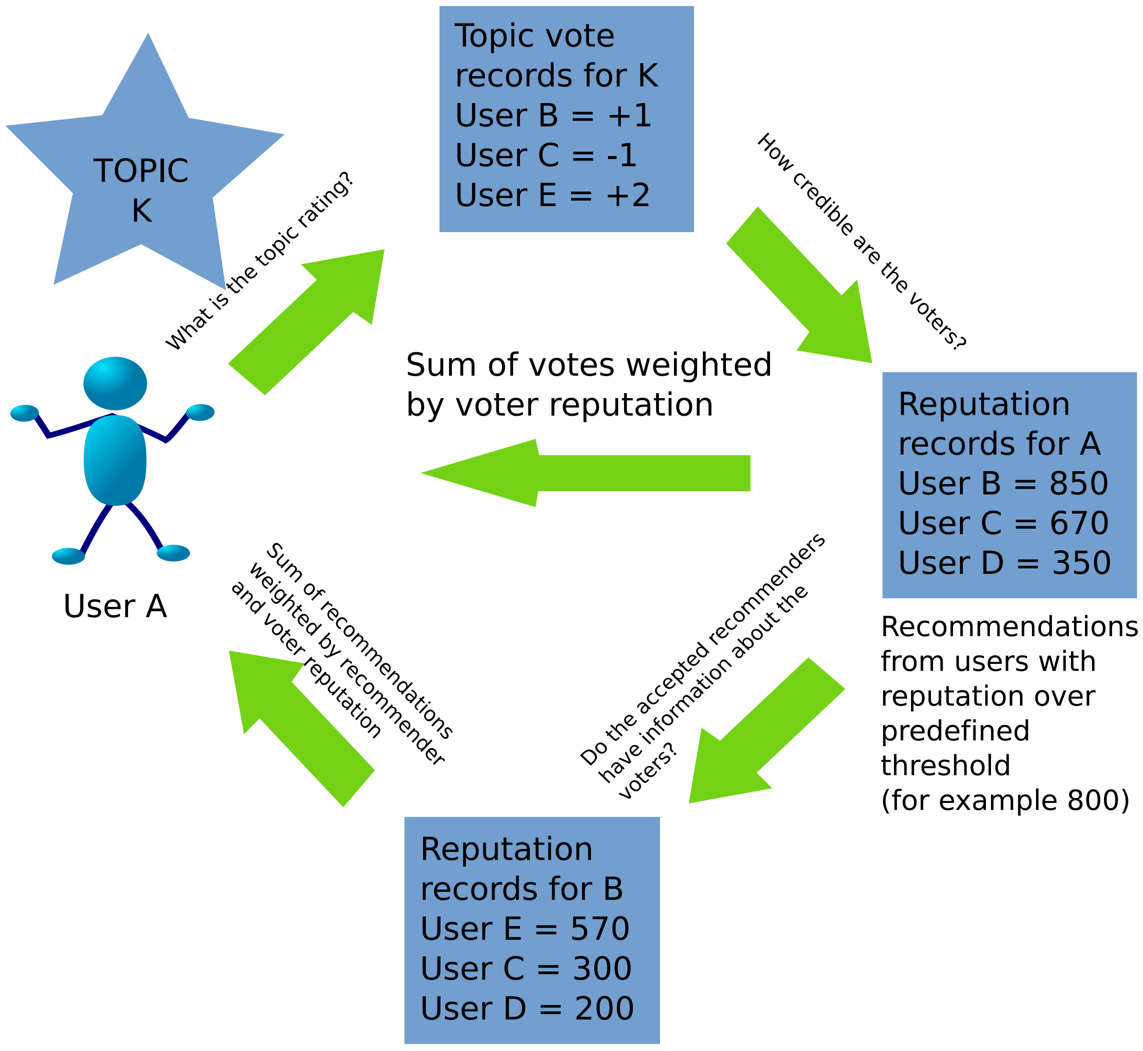

TOPIC K
Topic vote records for K
User B = +1
User C = -1
User E = +2
What is the topic rating?
How credible are the voters?
Sum of votes weighted by voter reputation
User A
Reputation records for A
User B = 850
User C = 670
User D = 350
Recommendations from users with reputation over predefined threshold (for example 800)
Sum of recommendations weighted by recommender and voter reputation
Do the accepted recommenders have information about the voters?
Reputation records for B
User E = 570
User C = 300
User D = 200

Figure 4.1: Topic rating

In JOP individual reputation for users was considered to be more useful and accurate, as the portal was intended to have different scopes and levels of expertise. One might be reputable among their peers due good edits on basic level topics, but this does not make one an expert in more advanced topics. Similarly, expertise in some specific field does not necessarily translate to expertise in a completely unrelated field. Figure 4.1 depicts the rating calculation process for a topic, when individual reputation scores are used with recommendations. Vote information in Topic vote records in figure 4.1 is weighted according to the reputation of the voters. The reputation information is kept in user specific reputation records. In addition, the reputation information from trusted peers is used to calculate a recommendation based score which is added to the total (Figure 4.1).

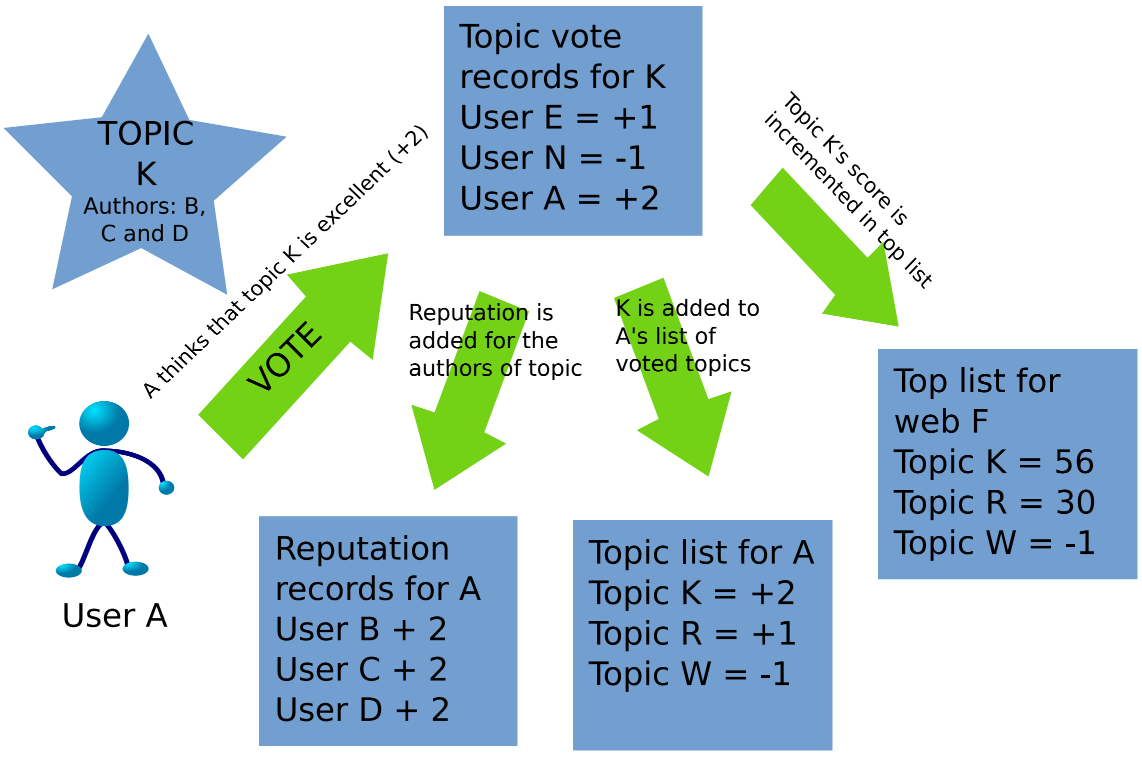


Figure 4.2: Topic voting

Reputation information is gathered from the individual votes (Figure 4.2). Because individual reputation has performance and rating quantity related downsides, the global reputation model was supported in TWiki-web specific aggregate listings. The records used to form the aggregate list is depicted by the Top list block in figure 4.2. These records are updated for every vote. In addition, user specific records for votes are kept in order to have user specific aggregate listings. These records are depicted by the topic list block in figure 4.2.

4.1.4 Normalizing Reputation

In order to have an easily understandable measure for topic quality, a normalized score calculation method was developed. The mean of the votes was calculated and this result was normalized to the range of 0..1 by dividing with the maximum vote value.

$$\frac{1}{n}\sum_{i=0}^{n-1} R_i \tag{4.2}$$

Biggest problem with using the mean value is that the topics which receive only few uniformly positive votes appear to be better than the topics with more votes but more versatile feedback [7, pp. 58 – 60]. This was the reason to display only the sum of the votes in the original implementation [9]. The problem was avoided by introducing neutral valued phantom votes to the calculation. This is a simplified version of the so called Bayesian average (Equation 4.3), which has both N_m, as it has the mean number of ratings and R_m the mean value of a rating in the whole set of rated items, set to a constant. Value of 0.5 was used for the N_m and zero for R_m (Equation 4.4). Due to rounding the phantom votes will become insignificant after fifty votes. The amount of phantom votes was made to be adjustable, and a good value will depend on the site's size.

$$\frac{N_m R_m + \sum_{i=0}^{n-1} R_i}{N_m + n} \tag{4.3}$$

$$\frac{\sum_{i=0}^{n-1} R_i}{0.5 + n} \tag{4.4}$$

If true bayesian average would be needed, it would be possible to calculate N_m and R_m from the existing votes, but the experimental value can work reasonably well and besides being simpler and more efficient to implement it can avoid being a high barrier for new content in mature communities.

4.1.5 Making Trust Decisions Based on Reputation

Because the saved reputation information was just the sum of positive and negative encounters, a function resembling the author's intuition about reputation based trust was created ad hoc. It was based on an idea that we can acquire small initial trust through relatively few interactions, but in order to trust more a considerable number of interactions is required. More scientific approach could have been used, as is suggested in Mui L., et al. research about computational models for trust and reputation [41], for the calculation of this value, but no applicable models were found.

The resulting nonlinear function with output values changing faster at the middle and slower at the ends of the range is presented in the equation 4.5. In this equation $T_i(j)$ is the result of this function and $R_{i,j}$ is the reputation value assigned by i to j in the reputation database. In other words, the trust function maps the reputation values to a range of $-1..1$ in which -1 means completely untrusted and 1 completely trusted. The default reputation assigned to newcomers is R_{def} and the maximum reputation is R_{max}. The equation is symmetric around R_{def} and the result of the sgn operator determines whether the result is positive or negative. The functions were chosen because they were available in Perl without any additional libraries. Keeping separate reputation information about different types of previous encounters would have given an option to use more complex and possibly more accurate formulas for determining the reputation on the multiplicative scale.

$$T_i(j) = \mathrm{sgn}(R_{i,j} - R_{\mathrm{def}})\sqrt{\sin \frac{|R_{i,j} - R_{\mathrm{def}}|}{R_{\mathrm{max}}}\pi} \tag{4.5}$$

The reputation based scores were normalized to percentage scores in the same way as the global ratings. The sum of the weightings given to ratings was used as n in the same simplified version of the Bayesian average (Equation 4.4).

4.1.6 Recommendations

Recommendations are gathered only from users who are considered trustworthy. The trust decision for accepting recommendations is based on the recommenders reputation score and on the defined reputation score threshold. This threshold for accepting recommenders can be freely defined by the user and even set to zero. Recommendations can be either positive or negative in value. Negative recommendations are negative reputation statements from trusted users. The use of negative recommendations is a necessity since the recommendations are cumulative over all the recommenders, and leaving them out would introduce a positive bias. Recommendations are summed as shown in the equation 4.6. In this equation $T_i(j)$ is the trust function output of user i's trust to recommender j and $T_j(k)$ is this recommender's trust to the voter k. The sum of the recommendations is then multiplied by the vote options numeric value and added to the topic's score.

$$Rec_{tot} = \frac{1}{n}\sum_{j,k} T_i(j) \times T_j(k), \text{ if } T_i(j) >= \text{recommendation threshold} \tag{4.6}$$

We can see from the equations 4.6 and 4.5 that if recommendations are accepted from recommenders with a reputation score lower than the default unknown value R_{def}, they have a negative value, according to the recommender's assumed unreliability. This behavior is not usually desired as one can not make the assumption that untrusted users behave exactly in an opposite way from trusted users, but rather that they are unreliable sources of information. For example, untrusted user defining some other user untrusted should not mean that we should trust that other user. In addition, unknown users are not used as recommenders, as the weight of their recommendations would be zero.

The equation 4.6 assumes that there is transitivity for reputation. However, the transitivity assumption may not always hold, and users of the system might not expect that behavior. Therefore the default threshold for accepting recommendations was set to 800. This limit requires at least 300 positive interactions with an user, which means that considerable amount of interaction or a deliberate change of either the trust value or the threshold is needed.

The recommendation chain was set to have a maximum length of only one step, meaning that only the directly known peers are used as recommenders. Longer chains would not provide as reliable information as direct recommendations, and the sources for information would be more difficult to manage. In addition, including only recommendations from directly trusted peers had performance benefits as well as a simpler implementation.

4.2 Trust Management Features

One of the important questions in this chapter is that: are the trust and reputation management ideas suitable for a wiki environment at all? Wikis implement the ideas of soft security by social control, but this approach can fail when encountering unknown entities which are malicious and possibly automated vandals such as spammers. Finding suitable partners by their reputation fails because the potential partners have to be allowed into the restricted areas where the abusive users can make substantial damage and if they are not allowed then there is no way for the users to contribute and gain reputation.

This consideration led to designing an out of band platform for partially untrusted collaboration. The main benefits of such a system are identifying reputable identities and allowing limited collaboration securely. Abuse was made easy to mitigate, as reputation based feedback filtering could be built to be an integral part of the system. The aim is that users who do not have write access to a topic can still contribute, and can do so safely without the possibility of including active content.

The earlier TWiki comment system implemented by the CommentPlugin [42] supports writing comments to a different target topic, and the topic containing the comments can be included to the current topic. These comments submitted through the CommentPlugin are filtered to remove any potentially unsafe characters. While providing commenting access without write access to the content itself, this model did not support partially untrusted collaboration as the target topic could be modified through the normal edit functions, and the result would get included to the current topic. Possible modifications could include inserting malicious material and changing the content of previous comments. More specific examples include link spam, which is common in unrestricted blog comments and javascript based attacks as TWiki does not prevent injection of client side code [43]. In addition, the comments were not treated as separate entities and could not be included individually.

Only a small patch was needed for the CommentPlugin to give the comments a unique identification number and a Hash Message Authentication Code (HMAC) for integrity. The HMAC uses a shared secret known by the CommentPlugin and the ReputationPlugin, which prevents others without the knowledge of the secret from computing a valid hash digest for a comment. When comments are displayed, ReputationPlugin will include only those comments that have a valid HMAC. The shared secret for the HMAC was devised to be configurable by the system admin, but it was also made to auto-configure to a 128-bit random value if custom value is not provided. This method provides a strong default password, while still maintaining the ability of transferring comments from a previous installation. Some protection against replay attacks was provided by including the topic name and the comment author name in the hash digest calculation. However, the comments can still be multiplied and reordered inside the same topic.

Moving the commenting system out of the wiki style social control to the reputation based approach is not suitable for all cases, as it can lead to strong groupthink. However, in JOP the common use scenario for a moderated commenting system are course materials, which require strict control over editing but benefit from comment access for improvement ideas. In such cases, the comment voting can be used to measure the popularity of the presented ideas.

4.3 Technical Environment

The production server, the web server where JOP was installed, was running on the host *jop.cs.tut.fi*. This host was part of the Lintula computer system, which provides Unix and Linux services for the Faculty of Computing and Electrical Engineering in TUT [44]. The Operating System (OS) was SunOS 5.10 (Solaris 10) and the actual portal was running inside a separate Solaris resource container (commonly referred as Solaris zones). Only user level access to this machine was granted by the Lintula administration, so it was not possible to restart services, such as Apache HTTP-server [45], or install additional software to system default locations. Additional Perl modules could be installed to user directories from CPAN, which is an archive of installable Perl modules [46]. The installed TWiki version was TWiki version 4.2.4, with TWiki plugin API version 1.2.

Because the JOP portal had been exhibiting slow response times the Apache module which provides cgi-environment had been changed from more common mod_cgi to mod_perl [47]. Mod_perl differs from mod_cgi in that it compiles the Perl code only once and retains it in memory, instead of compiling used scripts for every dynamic page request [48]. While the use of mod_perl made the portal more accessible to users, it created additional administrative complexity, as every change in code and configure settings required a web server service restart to take effect.

Code changes were not tested on the production environment, and a separate development environment was used instead. The development environment host was *huuliharppu.atm.tut.fi* and the OS was Ubuntu Linux 8.04.2. This host was not accessible outside of the TUT network. Root level access to this system was provided by the DCE administration, which made even large system configuration changes possible. Three separate wiki environments were installed on the development host. Two TWiki installs with different versions, the first was TWiki version 4.1.2 which was installed from Ubuntu software repositories for earlier TWiki related projects, and the second was more up to date TWiki version 5.0.1 installed from the TWiki subversion (SVN) repository [49] during this project. Third test environment was installed for compatibility with Foswiki [50]. Foswiki is a separate branch of TWiki maintained by a group of former TWiki developers. Foswiki compatibility was considered important as its development seemed more active than TWiki's, [51] and there was a possibility of JOP-portal moving to this platform [1].

4.4 Internal Data Structure

The first version of ReputationPlugin used comma separated text files for storage [9]. While it was a computationally efficient and dependency free solution, it added much complexity and inflexibility to the code. Parsing text required a lot of additional error checking, and the results had to be converted to structural data constructs for most operations. In addition, small text files are not efficient in terms of disk space, as they reserve a small additional overhead depending on the filesystem. Because the stored records were very small this overhead was much larger than the space needed for the information itself.

To address the issues concerning flat text files, a switch was made to MLDBM [52] and BerkeleyDB [53]. BerkeleyDB is a fast embedded database, which supports simple key value pairs. BerkeleyDB was used as the storage engine for serialized databases and the serialization was done with MLDBM. MLDBM is a CPAN module which offers an easy to use way to store Perl internal data structures to a flat file or a database. The change was driven in part by the need to change the previous record format. For example, the revision information was missing and to include the revision information efficiently a complete redesign of the record format was necessary. A version of the plugin using comma separated flat files was included in the distribution package, but it was not developed further.

Four different databases were implemented, two of them were fully independent and two of them had a relationship between each other. The quota system, which was made to provide additional voting restrictions, was left to use flat files for storage as it lacked an administrative interface and the quota files may require manual adjustments which are fast to do with human readable flat text files. Users and topics had their own vote databases. These are databases are not independent and they can be built from one another with additional information from the running TWiki system. The topic-votes database is used when voting results are displayed. The user-votes database handles user to topic relationships and includes a list of credited authors needed for vote deletion. The two independent databases, reputation and top list database, aggregate reputation information about content and users for performance reasons. Because the reputation database is separate from the other databases it is possible to change these values freely.

There is a strong argument for using an external relational database engine, as it would provide built in support for some of the data storage and retrieval operations. In addition, the use of a relational database engine would reduce the need for duplicate data needed for implementation simplicity and performance reasons. Indeed, the current implementation uses a separate database for almost every action. Nevertheless, because there were no issues concerning the needed storage space, the current approach was seen as sufficient.

4.4.1 Topic Votes Database

Topic Votes Database (TVDB) is used to hold voting information to be used in displaying the ratings. The database structure is designed to make the most used function fast and simple to implement. Counting the votes is performed every time a logged in user views a voted topic. Revision information is included in the tree which makes it possible to count the votes according to a specific topic revision. When votes are added it is necessary to check whether a previous vote by the same user exists, and because the username is at the bottom of the hierarchy searching for it is relatively slow. However, voting and removing votes are not as frequent as counting and displaying the votes. Figure 4.3 shows the structure of the TVDB for a topic. In the figure the multicolumn structures are called hash constructs and the arrows lead to values from a key. In Perl the hash tables can be multidimensional ie. a hash table can have another hash table as a value.

The information in this database is supposed to be presented through the plugin as anonymous data. However, there are certain situations where this does not necessarily hold true. For example, if the votes are allowed to be counted by using reputation information a user can reveal any voters identity by trusting only the suspected voter and looking at the topic rating. This type of attack can't be prevented in the current implementation, without restricting the use of reputation information. A simple solution to this privacy issue would be to limit giving out content recommendations only to reputable users. However, as limiting the recommendations would add performance overhead while reducing the utility of the system it was not considered important enough to implement.

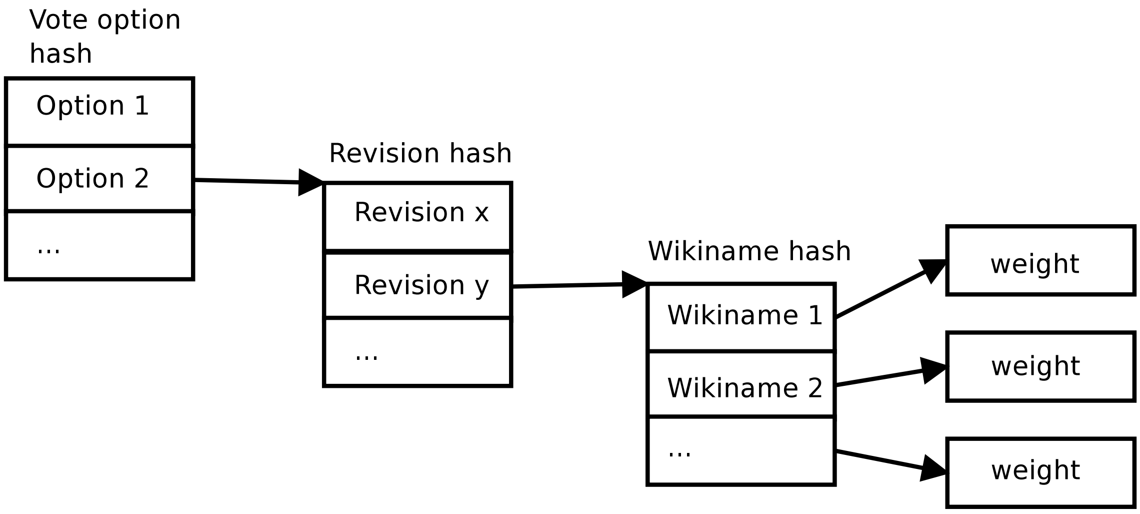


Figure 4.3: Data structure used to save vote information for topics

4.4.2 User Votes Database

User Votes Database (UVDB) was originally made to track the wiki names who were the authors of the topic at the time of the vote, as this was required for removing votes in an efficient manner. Every user on a TWiki site has a *login name* and a *wiki name*, in many installations these are the same. Login name is used for internal purposes, such as authentication, and wiki name is shown to other wiki users. This database can also be used to give a list of every topic a spesific user has voted without going through the whole TVDB. User's vote contains information about the voted topic and the voted topic revision, Wiki names of the authors of the topic at the time of the vote are included for performance reasons. This database is structured to have the unique key formed from the web and topic name as the only key. The data structure for one vote instance is shown in Figure 4.4.

Access to this database is limited so that every user can read only their own records through the plugin. Any function reading this database does not take any arguments which define the user, but instead the login name is taken from the TWiki's global user-variable.

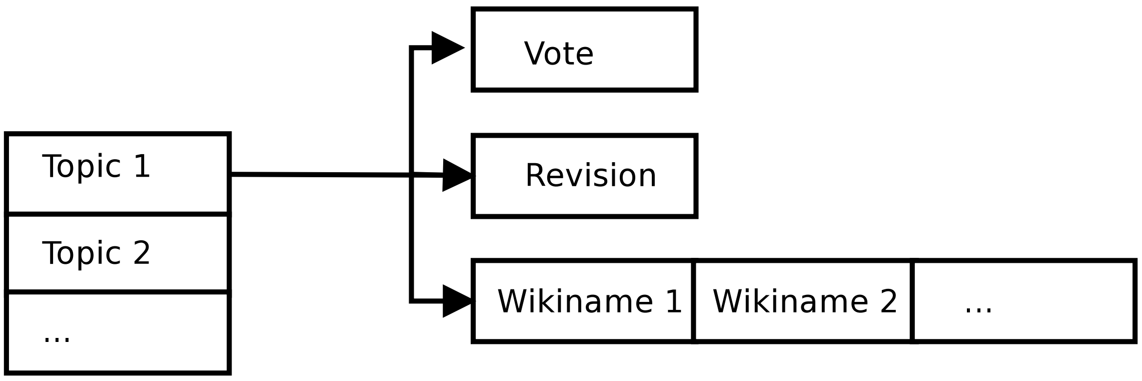


Figure 4.4: Data structure used to save vote information for a user

4.4.3 Reputation Database

Reputation database (RDB) was made to be separate from the UVDB to allow fast access to reputation scores, as using UVDB for this task would require going through every record in it every time reputation information for one or multiple users is needed. This arrangement had the added benefit of a possibility to change and add the values at will. The RDB file contains separate databases for all users. Database can be retrieved from BerkeleyDB with user's login name as the key.

The databases consist of simple key value pairs. The keys in user's RDB are unique Wikinames of the other users. The values are limited to a range of one to 999. Zero was left out for practical reasons as in Perl it is easy to be mistaken as undefined value. The range can be redefined by changing global constants in the code, if more or less granularity is needed.

The values in this database are not intended to be public, but these values can be extracted from the system under spesific circumstances. A user can find out a specific target's reputation score assigned to them by using an additional sock puppet account (sybil attack) and a topic with only one vote from the main account. Using a known value for reputation from the sock puppet account to the target account would reveal the reputation value in the recommendation received from the target account. This attack is possible only if the use of recommendations is allowed and the attacking user has vote privileges to a relatively unvoted topic. Method to mitigate this attack would be to allow the use of recommendation information by only the users which the recommending user trusts.

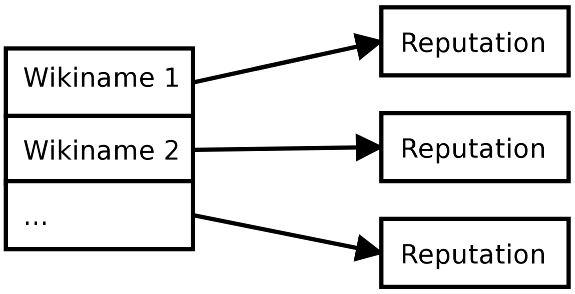


Figure 4.5: Data structure for reputation database

4.4.4 Top List Database

Top list database (TLDB) is used to aggregate vote information to TWiki-web specific top lists. This database is separate from all the others and it is not tied user reputation. However, the top list feature was considered to be something that the end users would like to have. Because it was a desirable feature and the time needed to implement it was neglible it was added. It would have been possible to implement this functionality without this database by using the TVDBs of all the topics in a TWiki-web, but it would have had performance drawback.

Separate databases are made for all the TWiki-webs. Inside each database there are two values for every topic: the sum of the votes and the count of the votes. True bayesian average was calculated for each topic in the list using equation 4.3. This calculation needed a special record that had the precalculated mean number of votes and mean vote value. The calculations were made for every toplist view. Performance would have been better if an approximate (equation 4.4) bayesian average would have been used with constant values for the mean number of votes and the mean rating. However, if performance would be a concern the best option would be to cache the results.

4.5 Settings and customization

Default settings were intended to be safe in the sense of anonymity and voting result integrity, but not overly restricting. The principle was that, the plugin and most of its functions should be usable without any additional settings. However, every value which was safe and practical to be configurable with a separate preference setting was made so. These settings are mainly intended for the site administrators, but some of them are also user configurable.

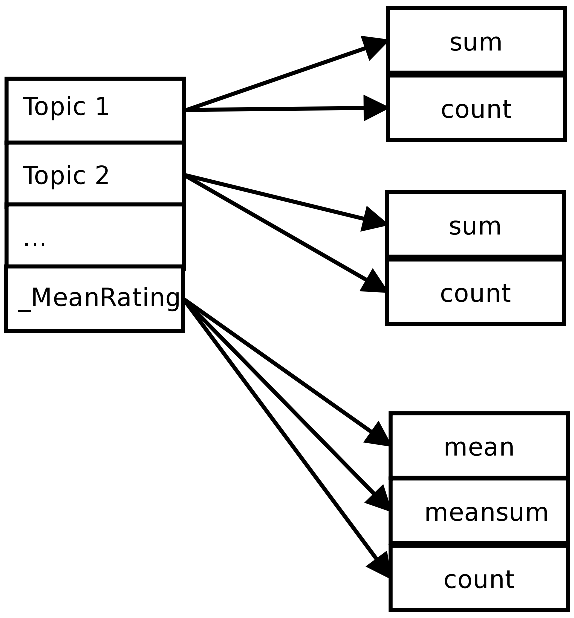


Figure 4.6: Data structure for vote summaries

TWiki has two kinds of variables for settings, the *configuration file variables*, and the *preferences variables* which can be set inside the TWiki system. Most of the settings were made to use the preferences variables, as setting the configuration variables required command-line administrative access to the host itself or access to TWiki's configure interface [54]. In addition to the requirement for administrative access, use of mod_perl [48] on the production environment required a web server restart for the configuration file variable changes to take effect [47]. Preferences variables have a downside as well. Because they can be set by regular users, additional care must be taken to protect their use.

The provided preference variables follow the TWiki access control rules in general [55], but some additional limitations were put into place as some of the settings are only applicable to a whole web, such as web spesific quota and the visibility of top lists, and allowing the setting of them in an individual topic could create undesired outcomes. Table 4.1 lists all the variables which have an effect only when used in a web specific preference topic. For example, the weightings for votes makes sense only when there is a group of more authoritative voters. One such scenario could be an exercise work for a course with the teacher having more influence on the vote outcome. Quota functionality, which restricts the amount of votes on a web basis, was implemented in order to have limited votings for cases where a couple of best topics are voted from a TWiki-web.

Table 4.1: Web spesific preference variables

Variable name	Type	Description
TOPLISTHIDDEN	Boolean	If set the web's toplist is not visible
WEIGHTGROUPS	List	Comma separated list of special groups and the weights they have in the ratings
WEBQUOTA	Boolean	Boolean option to switch on quota related functions for a web
QUOTATMPL	List	Comma separated list of vote options and the maximum amount of votes for each option.

Access control variables are in Table 4.2. These are comma separated lists of users and groups. Since these variables are standard TWiki access control variables, it is possible to use deny lists as well. These settings are independent of each other in order to make blind voting possible. In the blind voting mode a user cannot see the votes of others, but is still allowed to vote. Table 4.3 lists the other preference variables, most of these end user configuration, although it is possible for the local TWiki administrator to provide custom site wide default values, and even to lock them as final preferences [55].

Table 4.2: Access control variables

Variable name	Type	Description
ALLOWTOPICRPREAD	List	List of users and groups who are allowed to view the submitted ratings
ALLOWTOPICRPVOTE	List	List of users and groups who are allowed to vote

Table 4.3: User, topic and web configurable preference variables

Variable name	Type	Description
VOTEOPTIONS	Number	Number of voteoptions shown and allowed
RATINGTYPE	String	The type of rating calculated for topics. This can be “relative” or “absolute”
TRUSTTHRESHOLD	Number	Minimal value of reputation to qualify as a rating source. The default value was set to the default value 500.
RECOMMENDATION-THRESHOLD	Number	Minimum value of reputation to qualify as a recommender. The default value was set to 800.

4.6 User Interface Parts and their Functionality

The user interface consists of five separate task oriented parts. The main interface is the voting interface that is shown every time a logged in user views a topic, which has the plugin enabled. The comment interface is intended to be used with the main interface on the bottom of every topic, although both can be included separately as well. The three other interfaces are helper interfaces for additional tasks such as changing the reputation scores, viewing user's own votes and using the web specific top list functionality. These helper interfaces were implemented as separate topics.

4.6.1 Voting Interface

Figure 4.7 shows the voting interface in its default mode. This interface's main design objective was to be simple and user friendly. Emoticons are used as a visual cue, and to invite the user to interact. The topic's rating is displayed first and the distribution of votes is visible in the buttons themselves. The ratings were showed as either raw scores or percentages. They were included as regular text, although additional styling would have gathered more attention. Optionally the scores can be displayed as a line of stars if the RatinContrib [56] module is installed. When a user presses the voting button, the vote is registered and the button changes to a remove button in the user's view.

The number of options was limited to two in default mode to save screen space. This behavior is suitable for situations where the use of this plugin is not in a central role or the votes are approving votes by their nature. If more granular feedback is needed, the voting interface can be customed with a TWiki preference setting *VOTEOPTIONS* presented in Table 4.3 to use more voting options. Currently supported modes are two and four options, shown in figures 4.7 and 4.8.

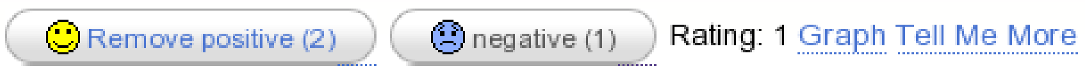


Figure 4.7: Voting interface with default settings

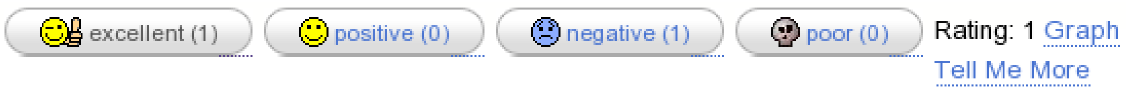
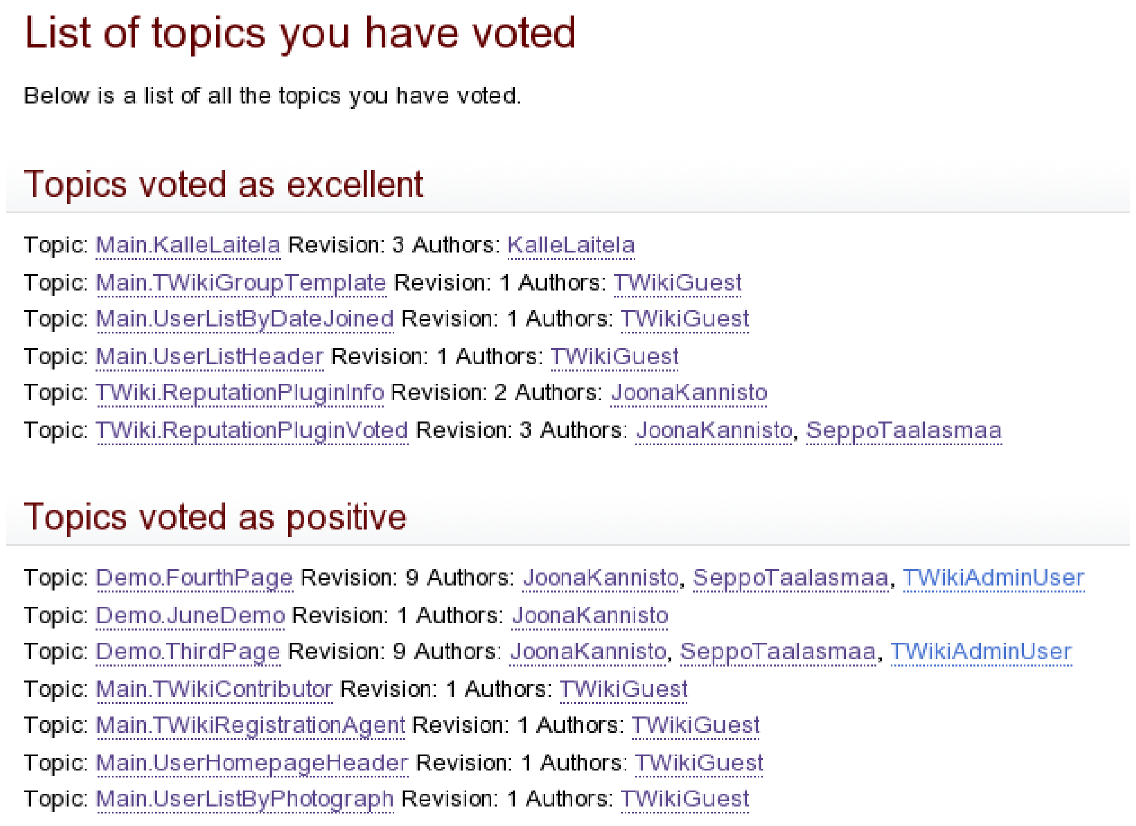


Figure 4.8: Voting interface with four voting options

Besides the buttons, there are two links in the interface, the first is a link that shows the vote distribution in a graphic form, this link is present only when the topic has at least one vote. The second *Tell Me More* link is a help function, leading to a topic where the features of the plugin are described. The separate help page was created to inform interested users about the plugin's operation and additional capabilities without presenting this information actively in the interface.

4.6.2 Vote Information Listings

To reward users for active participation, a special topic listing all of the topics that the user has voted was created. Indeed, this listing can be used as a bookmark feature for topics that are interesting and for topics that require improvements. The list is sorted in descending order according to the given votes. The voted revision and the authors are shown in addition to the topic name. Each topic's name is a link to the topic in question, and the authors' names are links to authors' personal topics.


List of topics you have voted

Below is a list of all the topics you have voted.

Topics voted as excellent

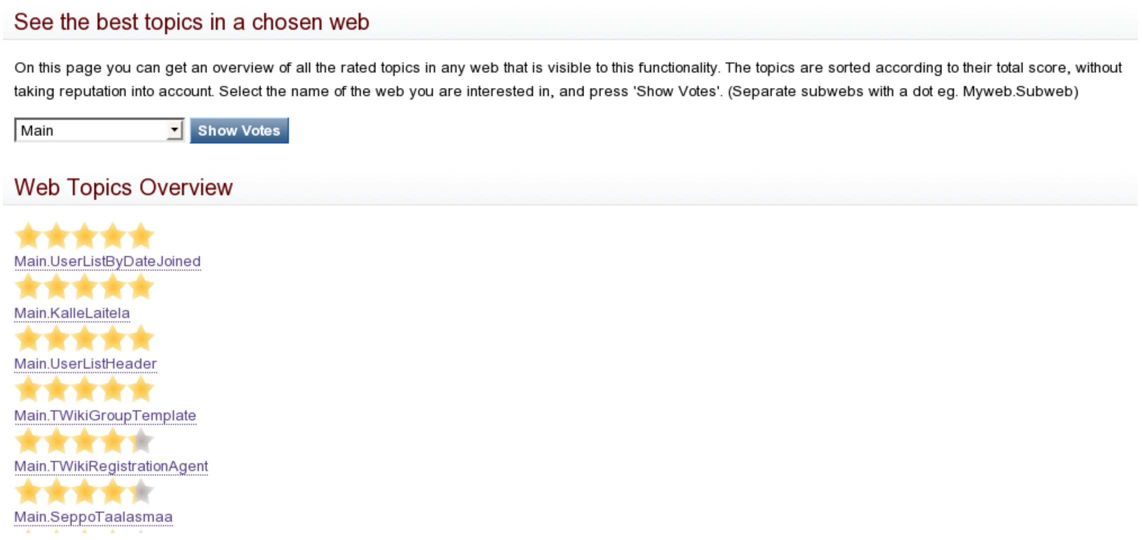
Topic: Main.KalleLaitela Revision: 3 Authors: KalleLaitela
Topic: Main.TWikiGroupTemplate Revision: 1 Authors: TWikiGuest
Topic: Main.UserListByDateJoined Revision: 1 Authors: TWikiGuest
Topic: Main.UserListHeader Revision: 1 Authors: TWikiGuest
Topic: TWiki.ReputationPluginInfo Revision: 2 Authors: JoonaKannisto
Topic: TWiki.ReputationPluginVoted Revision: 3 Authors: JoonaKannisto, SeppoTaalasmaa

Topics voted as positive

Topic: Demo.FourthPage Revision: 9 Authors: JoonaKannisto, SeppoTaalasmaa, TWikiAdminUser
Topic: Demo.JuneDemo Revision: 1 Authors: JoonaKannisto
Topic: Demo.ThirdPage Revision: 9 Authors: JoonaKannisto, SeppoTaalasmaa, TWikiAdminUser
Topic: Main.TWikiContributor Revision: 1 Authors: TWikiGuest
Topic: Main.TWikiRegistrationAgent Revision: 1 Authors: TWikiGuest
Topic: Main.UserHomepageHeader Revision: 1 Authors: TWikiGuest
Topic: Main.UserListByPhotograph Revision: 1 Authors: TWikiGuest

Figure 4.9: Listing of all the votes given by the user

A quick look into ten best rated topics inside a web was provided with a top list functionality. The topics were sorted according to the their calculated Bayesian average (Equation 4.3). The displayed amount of topics was limited to ten and only the best topics were shown. The topic names were included as links and the scores were shown as either percentages or as a line of stars. For this feature, a separate topic was made to the system-web. The system-web is a special web containing topics related to the platform itself. In this separate topic, the top list for a desired web could be accessed by selecting the web name from the select element on the page.


See the best topics in a chosen web

On this page you can get an overview of all the rated topics in any web that is visible to this functionality. The topics are sorted according to their total score, without taking reputation into account. Select the name of the web you are interested in, and press 'Show Votes'. (Separate subwebs with a dot eg. Myweb.Subweb)

Web Topics Overview

Main.UserListByDateJoined
Main.KalleLaitela
Main.UserListHeader
Main.TWikiGroupTemplate
Main.TWikiRegistrationAgent
Main.SeppoTaalasmaa

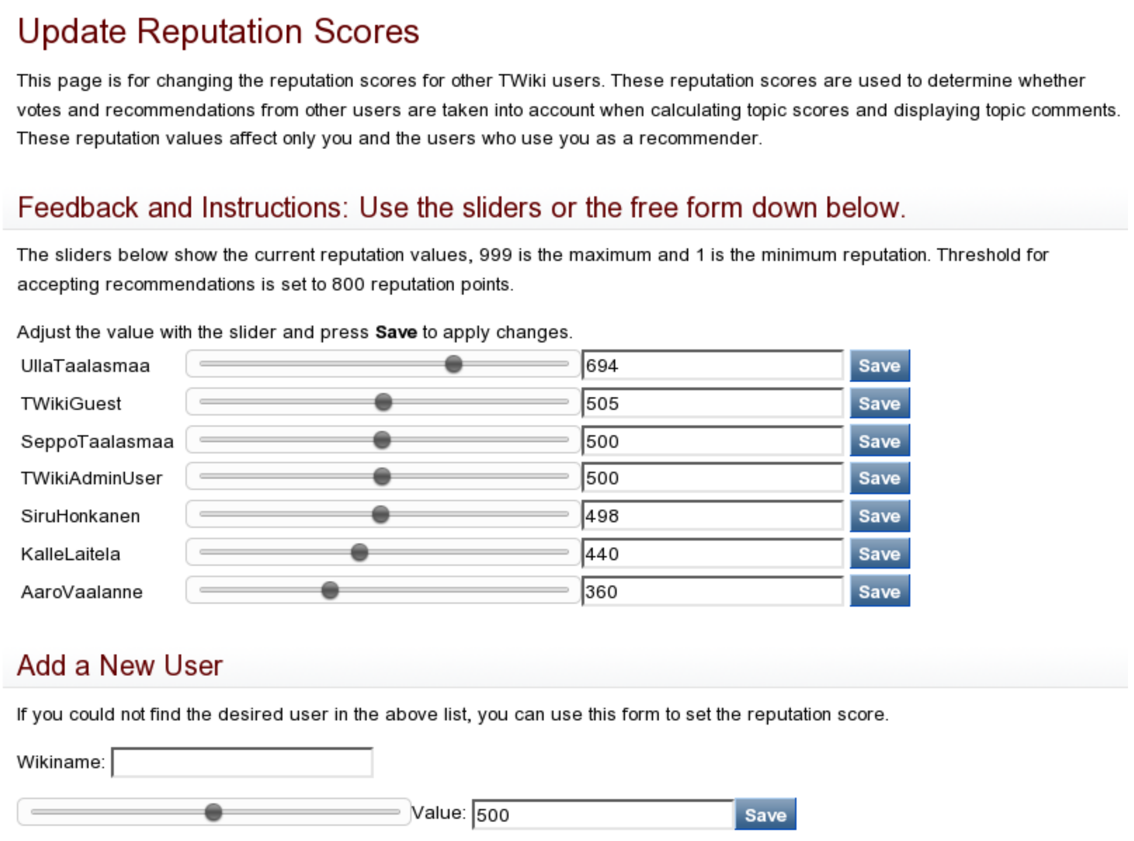
Figure 4.10: Best topics in the web ”Main“

4.6.3 The Reputation Score Interface

Reputation scores were made entirely user configurable as the users might have relationships created outside the online environment. The configuration dialog was put to a separate topic, where the plugin was called with the needed parameters. With these parameters the plugin displays the reputation score change form. Current values were shown in the sliders. The plugin was set to create the HTML for the sliders and the free submit form was in the topic itself (shown in Figure 4.11).

The user uses either the slider or the textual input beside the slider to change one user’s reputation score to a desired value and submits the change by pressing the **Save**-button located at the end of the slider. Only one value can be changed at a time. If there has been no previous interaction and the desired user is not in the RDB, their wiki name has to be given in the form at the bottom of the page.

In addition to users who utilize the reputation based scoring system, this view is most important to users who are in the position to grant access privileges. As it is possible to see with a quick glance which users are the best possible collaborators.



Update Reputation Scores

This page is for changing the reputation scores for other TWiki users. These reputation scores are used to determine whether votes and recommendations from other users are taken into account when calculating topic scores and displaying topic comments. These reputation values affect only you and the users who use you as a recommender.

Feedback and Instructions: Use the sliders or the free form down below.

The sliders below show the current reputation values, 999 is the maximum and 1 is the minimum reputation. Threshold for accepting recommendations is set to 800 reputation points.

Adjust the value with the slider and press **Save** to apply changes.

UllaTaalasmaa	694
TWikiGuest	505
SeppoTaalasmaa	500
TWikiAdminUser	500
SiruHonkanen	498
KalleLaitela	440
AaroVaalanne	360

Add a New User

If you could not find the desired user in the above list, you can use this form to set the reputation score.

Wikiname:

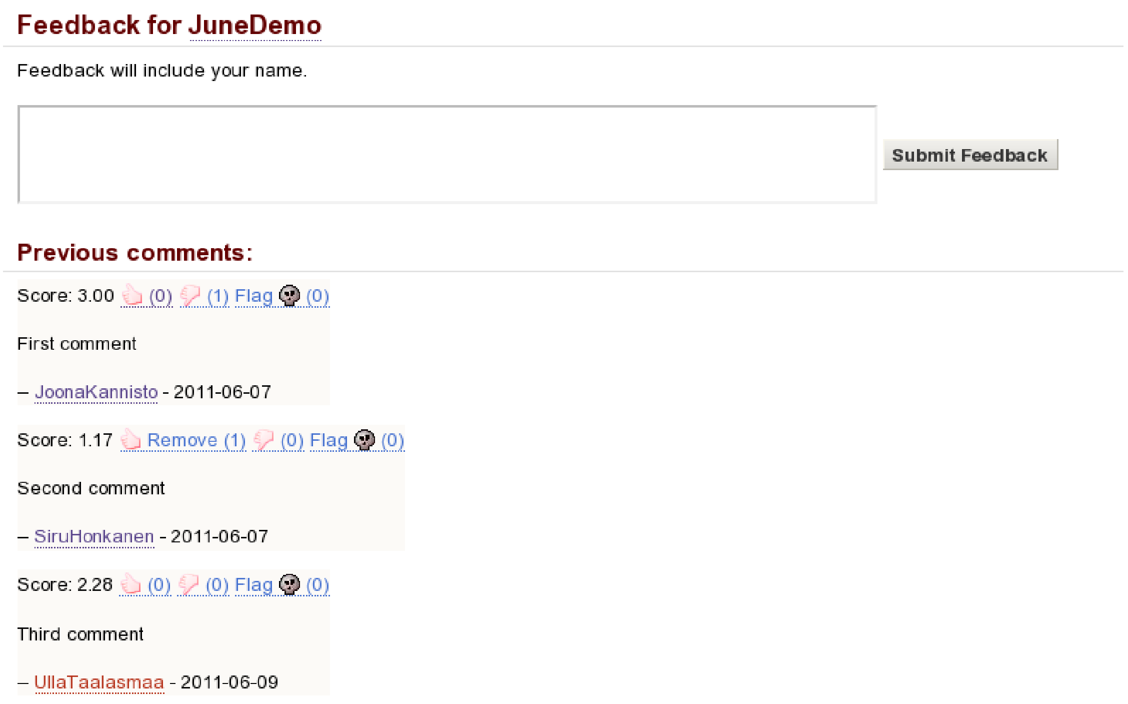
Value: 500

Figure 4.11: Reputation score change form

The reputation score changing functionality was restricted to be accessible only from the provided template topic by default, as a malicious user could include the command in any topic in an uninteractive mode, possibly as a HTML-comment, and change the visiting user's reputation scores without their acknowledgement.

4.6.4 The Comment Interface

The commenting interface (Figure 4.12) uses modified TWiki extension Comment-Plugin to provide the comment submitting functions. The existing comments are included through ReputationPlugin selectively with a slight shade in background color. If a comment does not have enough votes or the author of the comment is unknown the comment is displayed in a collapsed form. Comments which are voted down by reputable users or have an author with a bad reputation, are removed from the view altogether.


Feedback for JuneDemo

Feedback will include your name.

Submit Feedback

Previous comments:

Score: 3.00 (0) (1) Flag (0)

First comment

– JoonaKannisto - 2011-06-07

Score: 1.17 Remove (1) (0) Flag (0)

Second comment

– SiruHonkanen - 2011-06-07

Score: 2.28 (0) (0) Flag (0)

Third comment

– UllaTaalasmaa - 2011-06-09

Figure 4.12: The comment interface with three user comments

Voting comments could be done directly from this interface. Every comment has small added thumb buttons for voting the comments up or down. There is a special button to mark the comment as offending, but as reporting functions were not implemented this button currently only provides a more powerful downvote.

5 FIELD TEST

A field test for the ReputationPlugin was set up on the target platform. The studied things were the implementation's technical specifics rather than a test of the reputation framework, as the testing of the reputation management functions would have required much more free interaction between the test participants. The field test had six separate phases:

1. The plugin was installed to the JOP-portal

2. Instructions for the assignment were written

3. The plugin was configured to meet the specification

4. The students did the course work, including the peer review

5. Examination of the voting results

6. Feedback from the students

ReputationPlugin was used in a peer review part of a course assignment for the Advanced Course in Information Security. The main course assignment included moving and updating course material from a previous portal to the JOP. These study material topics were published in the separate Tietoturva web (Information security web) [17], which was publicly accessible. Besides these study material topics, the students were required to create a personal reflection topic to a course specific web. This topic was a diary of the work completion and a reflection about the topics they had chosen. The test setting was limited in order to track participation and limit the access to the students on that course. Access right and quota functions were implemented to the plugin for this purpose. Two different voting scenarios were created, one which was without any special limitations and the other with restricted number of votes as well as additional instructions about the allowed target topics for every user. The votes were supposed to be given to two best and to two worst topics.

One of the students did a special assignment and was required to read all the topics and give text comments on them. This additional student had unlimited voting quota and he was instructed to vote as many topics as there were exceptionally good or insufficient quality topics.

No additional instructions for the voting tool itself were provided, as the intention was to model a regular use case where the users are not given any training. Additional training would have its place in a time restricted setting, where the subjects would not have time to explore the features by themselves. Also, the user interface was intended to be self explanatory, and efforts were concentrated on its familiarity and clearness instead of supplemental documentation.

Feedback from the students was collected through mandatory course feedback. This feedback included user impressions about ReputationPlugin. The feedback along with first hand deployment related information was used to improve the plugin.

In addition to functional testing and user experiences, voting activity and bias in the target environment were of interest as well. For voting activity the hypothesis was that the students would not exercise their right to vote more than the specified minimum amount. The bias hypothesis was that the votes for the personal reflection topics would be much more persona centric and as such they would exhibit a strong positive bias, similar to the bias in eBay ratings [57]. There might be signs of social aspects such as retaliation as well. In contrast, the votes for the content topics were thought to be more balanced and not related to personal relationships as all the participants were required to give both positive and negative reviews and the subset of topics to review was defined in advance. However, it was interesting to see whether the participants withhold giving negative reviews, even though they were required to give them.

5.1 Deployment

As the target platform was different from the development environment, it was necessary to test the functionality, before the actual field test took place. During the test, the plugin did not depend on any additional Perl-modules, not included in Perl version 5.8.4 or in the TWiki base install, and the deployment was done by just copying the relevant files to the target server. The use of mod_perl instead of mod_cgi was the only difference between the test and production servers that had any impact.

Problems were encountered due to mod_perl, and fixing them was slow because of the need for server restarts. However, in the end only minor changes to the code were required. For example, mod_perl makes global variables, which are sometimes used for settings, persistent over sessions, so the default values for these had to be assigned in initialization routines.

5.2 Feedback about the Voting Tool

The answers were originally in Finnish, and they have been translated into English. The intention with the translation was to preserve the meaning as well as the tone of the responses.

The first question was: “Did you encounter any problems with the voting tool in the third stage”, which was referring to the peer review stage of the assignment. As the question was especially targeted at finding problems most of the feedback was centered around that, rather than any general feedback. Nine short answers, which stated in few words that they did not experience any problems or shortcomings are not included below.

One student complained about lack of feedback in the voting tool and unclear instructions. This claimed lack of feedback should not have been possible, since the plugin provides textual feedback for every action. It is possible that the feedback was not visible enough, or that the student had a browser compatibility problem which prevented them from interacting with the plugin altogether.

> “The voting tool felt a bit complicated, as it did not give any notification whether the vote was registered or not. In addition, the instructions for its use could have been clearer.”

One functional problem reported by three students was discovered, namely topics made by one author could not be voted. This author had used a colon in the topic name, and as a result of this, a malformed target url for the voting link was created by the plugin. TWiki tries to prevent the use of non alphanumeric characters in topic names by filtering them in the *Create New Topic* -page, but it does not enforce this rule in the actual topic creation phase i.e. it is possible to manually type an *URL*, such as *http://jop.cs.tut.fi/twiki/bin/edit/Sandbox/TopicWith:Colon* and save the topic through the edit script. The resulting problem was fixed by percent-encoding [58] colons, as well as other reserved characters defined in RFC 3986 [58], in returned URLs.

> “All the topics could not be voted. I commented this on my reflection topic: ‘For some reason I could not vote topics with names starting with IPSec:’. I wonder whether anyone else has the same problem...”

> “Topics starting with Ipsec: could not be voted at all. Probably a problem with the links. I did not discover any additional problems related to voting.”

A usability problem was discovered, one of these three students complained about the quota functionality not showing the number of remaining votes, which was an obvious shortcoming. The reason there was no feedback about the quota related functions in the interface, was that the quota functionality was invoked only during the voting action for performance reasons.

> “Only problem I found was that certain topics could not be voted at all. One feature which I would have liked to have was to have the number of remaining votes to be visible somewhere. ”

Two students stated that they would have liked to have more options for feedback. The first would have liked to have a neutral vote, and the second was referring to vote options in general. Neutral vote was originally left out as it was considered unnecessary, if someone has the incentive to vote they are likely to have a strong opinion about the topic. This may not be true in all the scenarios, especially in this case where the users were required to vote.

> “One problem turned out to be the absence of a so called neutral option, as it was not always possible to say whether a positive or a negative option would be closer to the truth.”

> “In my opinion, the two-part scale of the voting tool was a bit unfitted for real page evaluation. Now it was only possible to tell which were the best and worst topics but there was no feel of evaluation.”

In conclusion most of the students (nine out of 16) had no problems using the voting tool. However, it is possible that some of these students were not motivated to answer in greater length, and avoided the question by denying any problems. In addition, it has to be noted that the feedback was written about a month after the field test took place, and some minor difficulties might have been forgotten. The features which were requested were larger scale for voting options and better feedback from the plugin.

5.3 Feedback About Peer Review

Another question, not strictly related to the voting tool but related to the peer review done with it, was asked from the students: “How did the peer review affect your perception about your own expertise?” The purpose of the question was to find out whether the students felt that the peer review capabilities were important to them. The perceived importance of peer review relates to the prevalence of egocentric recognition incentives [59, p. 53 -82] in the target audience and to the willingness to review others [7, 111-120]. The answers to this question were mainly about the textual feedback, which was at the time not handled through the plugin, although the question was intended to include both forms of feedback. While these answers did not mention the plugin as a review tool, they included insights about criteria for reviewing content. It would have been interesting to measure whether the students who got the best reviews from others regarded the peer review as more useful than those who did not get as good reviews. But because the answers were anonymous this possible connection could not be verified.

Fifteen students answered this question and the answers were mostly positive. Many students indicated that by reviewing others' work they had learned to evaluate their own work as well. About a third of the students (four) did not think that the peer review helped them in any way. It should be noted however, that the reviews might have been difficult to do because the task assignment was about moving and updating teaching material instead of creating completely new works. Due to this the main focus of these reviews was instructed to be on the improvements and refreshments made to the material. In addition, improvements were only necessary if the source material was unclear or had outdated information.

These answers suggest that peer review can give the students an understanding about their own strengths and weaknesses relative to others.

> "Peer review was completely positive, as it gave a clearer image about the quality of my work through evaluations from others, and through my evaluation of others' work."

> "It gave a more positive image than I thought."

> "Peer review brought out the successes and failures in the exercise works. It brought out things that would not necessarily have been noticed by oneself."

> "The effect was that I discovered that I knew even the content of the topics suprisingly well. Of course in every peer evaluation there is incentive to give positive feedback but through this course I learned to be more realistic."

> "It had a meaningful impact."

Five students got the feedback that they expected, this means either that the review process was considered to be fair or that the review did not reveal any new information.

> "I was in a prior agreement with those who reviewed me, so the review had no impact on me."

> “Peer review of the exercise works was of the constructive nature and mostly inline with my perception of my own expertise regarding updating material on TWiki. I did not fully understand what was the idea behind this question”

> “Peer review was of positive nature for the most, and it underlined the shortcomings which I already knew. It was not very useful as it did not go through the work (or the information within) profoundly, due to short lenght of the reviews.”

> “Hard to say. The feedback was what I expected, for the most. The reflection topic was successful in my opinion. Some of the phases in the assignment were left unfinished, and others were completed with slightly better results. Evaluating others’ topics gave me a comparison point to my own work.”

> “The feedback that I got was what I expected and in line with the work I had done. By comparing my work to others’ I would have evaluated myself in the same way.”

Few students found out through review that they had not understood the assignment, and for that reason had not done every required step. This problem could have been avoided by incorporating the peer review process in an earlier stage.

> “It made me realize that I had understood the assignment slightly wrong, and that I had not made necessary effort.”

> “Regarding the matter at hand, not much. I could have read the task assignments more thoroughly, so that I would have done every required step.”

Recognition and interest from peers was considered to be important by three students. This attention gives more meaning to the task as well, because it is possible to see from the reviews that someone has paid attention to their work.

> “Peer review brought up positively the fact that when there was substantial effort and the reviewer had had the time to read and think it all through, it could not have been made in few dozen minutes. Some personal weaknesses may have come out, and it is a good thing as it is possible to improve these things more in the future.”

> “It is always interesting to get feedback from one’s peers. The feedback that I got was well justified and very much what I expected it to be. As my input to this assignment was not full 100 %, I can’t use it as a mirror for my self image. In my opinion the work that I did was enough and mostly ok, and the evaluators agreed on that. That’s about it”

Quality of peer reviews might be an issue if the grounds for the review are not clearly defined. This answer stresses the need to have expert evaluation in addition to peer review, but using peer review can be useful for finding small errors.

> “The comments I got from peer review were great, and even clear errors were found because of them. Of course peer review may revolve around pointing at these details (finding errors), which is an easy way to accomplish one’s review task, while not working for the big picture in the same way as comments from a professional.”

> “The assignment was mostly about limiting the subject of the study, and finding information and how much time was possible use on it. I did not have much information about such specific topics beforehand. So, not that much.”

5.4 Results

The field test revealed several problems with the plugin, some of these were fixed during the test, but others were discovered only through user feedback after the test had finished. The problems which were fixed during or slightly before the test were mainly related to deployment, and the fixes were required to support the intended configuration.

Vote distribution for the content topics matched the expectation. The students did not withhold giving negative feedback, and the number of negative votes was slightly larger than positive ones. The larger number of negative votes was found to be due to the additional voter with a different task assignment from others. As expected, the votes for the reflection pages had a strong positive bias. Only four of the 50 votes were on the negative side, and two of them came from the additional voter who was required to rate each page. All the negative votes were given to two topics which were clearly unfinished. One explanation for the positive bias is that the reflection pages were more diary type of pages, and as such they did not contain any factual information which could be reviewed as true or false. The only discerning factor was that the pages were either done or not done and only the upper part of the voting scale was used to differentiate between the amount of work that had been put into the creation of the topic. This result suggests that in order to get better reviews, especially in the case that the reviews are personal, the scale has to be wider in order to differentiate between completed works of varying quality. One possibility would be to have , to provide more granularity to the positive side. [7, p.62]

In the test setting, the additional restrictions for the visibility, granularity and the amount of votes were reversed from review accuracy and incentive perspectives. The votes for the reflection topics should have been visible only for the topic's creator, as they can be considered to be personal topics. In contrast the content topics would have benefited from visible feedback as they are not persona centric and could easily be ranked. Amount of votes and voting time should not be restricted, because one of the peer review's strengths in teaching is that the review is educational for the reviewer as well, and the voting process for a course such as this should have been a multistage iterative process, with feedback possibility on every step.

Textual feedback was considered to be more valuable than voting as a peer review mechanism. However, the numerical feedback through voting did not achieve its full potential, partly because of restrictions. Numerical feedback concentrates on quantity instead of quality and it should be used to evaluate large number of topics instead of only few. Nevertheless, textual comments add clarity and additional information to reviews as the grounds for the review can be announced. A useful peer review tool should include text commenting in addition to a voting mechanism. Because of the strong preference for text feedback the plugin was later equipped with commenting features.

6 DISCUSSION

It is highly questionable whether content reviews can be used to evaluate trustworthiness or quality of an individual topic. Reviews can in some cases be centered around the popularity of the subject or the author. In addition, while content does have a reputation its trustworthiness is a more complex subject. An entertaining topic about an interesting theme, with some slight errors, should perhaps have a high reputation but its trustworthiness would not necessary be that high.

The wiki nature of the site makes the change tracking and evaluation of individual changes mandatory for an accurate system. Some of the desired accuracy was lost as topics were evaluated as whole. Topic evaluation was originally chosen as it requires less reviews than evaluating every single change. However, balance between the accuracy and the amount of input needed should have been configurable.

Perhaps the greatest single shortcoming of the plugin is the absolute requirement of user interaction. This requirement creates an obstacle for product acceptance especially in the bootstrapping phase. Other reputation inputs besides voting were considered in the original planning stage. Page views and ranking by the amount of inter wiki links (*backlinks*) leading to a topic were the considered approaches. Link evaluation was implemented, but as TWiki did not save the backlink information anywhere, counting these links required a whole TWiki wide search every time a topic was accessed. This would have created unreasonable performance overhead outside of small test systems. In addition, page views and *backlinks* are both measurements of impact factor and not quality or approval.

One reason user interaction was chosen instead of automated approaches was to eliminate possibilities of gaming the system. When reputation statements are only made knowingly the users are harder trick into doing them. However, the relevant question is whether automated systems fare better than systems which require user input in providing effective security. In order for users to use any system, the system has to prove its usefulness to the user.

Hybrid approaches would be interesting as well, possibly combining the content analysis, impact factor measured by the interwiki links and a subjective web of trust. Indeed, the best approach would be to combine all of these in a user configurable way. In such an approach the input from the automated system could be filtered as any other source of input, by using the information from the interactive reputation management system. Besides amount of data, additional metrics would have added more robustness [7, p. 74] to the system.

6.1 User Trust to the Reputation Management System

Trust to the reputation management system itself is an important consideration. Reputation management system's success is directly tied to whether the users find the information it provides trustworthy or not. This adds requirements mainly for transparency and usability. [27, p. 75] According to the model of trust by Egger F.N. [60], the origins of trust can be divided into three main factors: first is the appearance, the second is the ease of use of the system and the third is the information content from the system. Requirements for usability and appearance are similar to any other system or service, but the trustworthiness of the information content is different.

Users are more likely to trust the information when they have means to verify it. However, in the reputation management system presented in this thesis the identities of the voters were kept secret, which means that the users had no practical means for verifying the returned information. The lack of transparency was a result of a desire to have an anonymous system. Practical anonymity was desired as anonymous votes were thought to be less person centric and as such provide more objective feedback. Barrier of entry for anonymous reviews was considered to be lower as well.

The system offers only practical anonymity, which means that an attacker can confirm suspected voter usernames and reveal reputation information through the recommendation system. It is very difficult to have a fully anonymous system while still allowing interaction within the system through recommendations. An undesirable outcome is that the information can be extracted by officially unacceptable methods from the system by a malicious user, while the information stays out of reach for those who would have sincere motives but play by the rules. Because the anonymity was not accomplished by design, but instead by obfuscating the returned recommendations, neither strong anonymity nor transparency was achieved. If the desire would have been to provide strong anonymity, a separate option to cast anonymous votes should have been implemented.

Another target for transparency is the systems internal functionality. If the users understand the mechanics behind the system they are more likely to trust its evaluations. For example, by comparing the reputation management systems used on Slashdot and on eBay, we notice that the reputation management system on eBay is much more simple and understandable. The more complex system on Slashdot is developed for a different target group and it has a different threat model. While all the information exists, a user has to make effort to understand the operation of the system. In such case, the complexity creates a lack of transparency. Additionally, it is important how the required transparency is communicated to the user, as the user might not be interested in reading general help files, but would rather have something situation specific help. Such as a decomposition of a particular score on request.

The decision investment and associated risks affect the needed user trust. Systems which do not make automatic trust decisions do not need much trust from the users. For example, the reputation score of eBay reputation management system is not considered trustworthy by the eBay or its users. Indeed, the users are instructed to do manual evaluation from textual comments and the type of previous transactions [13]. However, this lack of trust does not make the reputation score useless, as it can be used to separate interesting peers from a large group of peers. Finding the interesting peers quickly saves time and makes it possible to concentrate the time consuming manual evaluation on the few interesting candidates instead.

One of the original goals [9] was to have the plugin provide access control functions based on the reputation of other authors. This was left out as it was considered that the risks involved would have out-weighed the benefits. However, this would have been a valuable tool for the users, as TWiki lacks easy to use mechanisms for access control for ordinary users. Group based access control can be out of reach and impractical, as ordinary users usually do not have rights to create groups at will, and the groups would have to be maintained manually. Listing individual users in topic access control rules is not practical. Users with administrative access could also benefit from the reputation based access control, due to its simplicity. Indeed, the plugin could be extended to provide a list of users above a given threshold in a compatible format for TWiki's access control variables. Nevertheless, the current variable evaluation order in TWiki makes it impossible to use the plugin's output as an input to access control variables in an ondemand fashion.

Solution which would require more work would be an agent which either approves, disapproves or places edits to a queue according to reputation of the editor in relation to the user who has placed the agent to guard the topic. This type of agent could be implemented without modifying TWiki core, as plugins do not fall under TWiki's access control. Read access could similarly be provided through an agent. Perhaps more widely used example of an agent utilizing reputation information would be one that would have access to all the reputation information for the purposes of removing abusive users. This agent would find malicious users by forming a web of trust from the viewpoint of a trusted root user and use a predefined threshold for negative reputation as the decision boundary.

The amount of reputation information available might create situation where the reputation information cannot be considered to be robust enough for large decisions. In order to utilize automatic decision making with the presence of such uncertainty the targets have to have small decision investment. The comments in the commenting system are an excellent example of this type of application. Reading a short comment takes about as much time as showing or hiding it through one click interface. This means that the decision to either show or hide a comment needs to be automatic or it will have no real benefit to the user.

6.2 Community Aspects

Important part in building a reputation management system is to think how it will best serve the community [7]. This can be problematic, when the reputation management system is built before the actual community has formed. The specific needs might not be known beforehand. Usually the process is iterative [7]. However, the foundations should remain constant as a constantly changing system can confuse users.

Keeping the reputation scores for users a secret was done to remove the competitive aspects of building individual reputation. Public reputation for content was considered to be a much better alternative, as it can not be accumulated by short-handed measures. If someone has a desire to be recognized they can produce content that will be rated high and use that content as a reflection for their individual reputation. Positive and negative user reputations are best to be used as an admin resource [7, p. 17]

6.3 Usability Considerations

The requirement for user interaction was not a bad thing from all perspectives. Interaction can have a positive influence on the system's familiarity to end users as well as for the goals of transparency. By interacting with the system the user feels in control of the sites information content, and knows where the content ratings come from. In addition, making users emotionally attached to a portal can work in small gradual steps: registration, then votes and comments on some topics, and finally edits to topics.

In the pilot test it was observed that every fifth user accessed the help page. While this number is substantial it is questionable whether it was acceptable to have some of the advanced functionalities behind this help page. One solution would be to implement these additional functions as well as the help pages themselves as drawer elements in the main user interface.

Threshold values for recommendations and rating sources were TWiki preferences, which meant that they had to be set by altering TWiki-topics. However, from the usability point of view these two values should be modifiable in the same interface as the reputation values. This would be intuitive and easy, and it would give a visible clue to what these settings actually do, as moving the bar higher would show which users are below it and which under. In order to accomplish this, these settings should be stored under reserved words in the RDB, as setting TWiki preferences was not supported in the Plugin API.

6.4 Ideas for Future Development

In order to have more versatile feedback, reputation inputs would need a wider scale, possibly one without explicit negative reviews. This requires a different interface, and some type of slider would likely be optimal. Amount of options depends on whether the votes are primarily peer review, or if they are approval votes from the content consumers to content producers [7, p. 62]. The most common rating scheme is star rating scheme and on the TWiki platform one could use RatingContrib. In addition to the positive ratings, there should be a separate flagging function as stars can be misleading if they produce negative feedback. The current four element scale was considered to be enough for the peer review case, and the two part scale for situations that are approval voting [7, p. 62].

One consideration about recommendations was whether they should have bidirectional properties, i.e. should the recommendations be different for different requestors. In addition, would it be desirable to have limits for giving out recommendations, as providing recommendations can be seen as a service to other users? At least from the privacy perspective, it should be possible to give out recommendations only to requests coming from the user's own trusted peers. This possibility was not considered in the initial planning stage as there was more pressure on promoting the spread of reputation information rather than limiting it. Some benefits for unidirectional trust relationships exist in the target environment. For example, a student might use a professor as a recommender for credibility evaluations of advanced information, without the requirement of interaction on this expertise level. However, the option to limit giving out recommendation information would be intuitive considering the social aspects, and it would enhance privacy.

Anonymity of the votes could be reconsidered. In the current implementation, the votes and the reputation information should not be thought of as anonymous data, as in the presence of recommendations their anonymity can not be guaranteed. In order to get rid of this disparity, one either needs to limit or remove the recommendations or remove the anonymity assumption of the votes. However, the lack of anonymity does not mean that the identities of the voters need to be public. Indeed, access to this information can be limited to the users whom the voter considers to be reputable.

The implemented commenting system leaks data in some circumstances as the target web for the comments may be set as public. If the commenting functionality is used in topics whose existence is confidential or the commenter identities or the comments themselves are considered confidential, some unwanted leaks may happen. Encrypting the comments and topic names would be an obvious solution, but it would remove every benefit that was offered by including the comments as TWiki topic text. In addition, because of the way the CommentPlugin and the TWiki revision history work, the comment author's name would be visible in the changelog even if the comments themselves are encrypted. If strict confidentiality is desired, the most reasonable choice would be to store the comments in a separate database.

7 CONCLUSIONS

The goal was to improve existing the reputation management system implementation for JOP, and to extend it to provide stronger trust management functions. There were substantive advances in usability and user friendliness. The reputation management system's implementation was tested in a small scale user test, and user impressions were collected about the appearance and usability of the system.

On the implementation side there were big changes to the previous software. New functionalities were implemented, and data storage was made simpler. Due to these changes, almost all the source code was rewritten. As the plugin source code was published as open source, an understandable and clean code base was essential for continuing development by the community.

In a wiki, poll driven content reputation can be used to measure content quality in cases where the decision investment is small. However, if the reputation has no strict accountability it will be exploitable for big decisions by malicious users. For this reason it was necessary to make a more easily accountable service for the TWiki platform. Topic commenting possibility, which stored the comments indivially and with integrity, made it possible to evaluate users and content reliably. This accountablity was necessary for trust management features, and automated trust decisions were implemented into the developed commenting system. In addition to small trust decisions concerning comment visibility, the commenting system could be used to establish trusted pseudonyms on the portal, while allowing a safe amount of collaboration by untrusted users.

The reputation management features were intended to be used for trust management as well as engaging users to form a community. Because reputation management systems provide a form of communication, they can have a strong influence on the community's behavior. These community aspects imposed some limits and needs for the displayed information, for example user reputation was not made public in order to limit competitive incentives. Furthermore, feedback by voting and secure commenting possibilities were valuable additions, as such communication channels did not exist on the TWiki platform.

It is now possible to evaluate content in JOP and use these evaluations to build user reputation. This reputation can be used to filter the content evaluations and to provide partly restricted access through a commenting system. The first implementation was a working prototype in order to see if something like this could be implemented, and this thesis extended the existing implementation with more value to the end users. Besides providing an already functioning and usable implementation, this thesis is likely to set future directions for reputation management systems on the TWiki platform.

REFERENCES

[1] Koskinen J.A. & Helenius M. 2010. Kehittyvän tiedon yhteisöllinen opetusportaali, JOP. [WWW]. [In Finnish]. [Cited 25.11.2011]. Available at: `http://jop.cs.tut.fi/twiki/bin/view/JOP/JopKuvaus`.

[2] Thoeny P., et al. 2011. TWiki - Open Source Enterprise Wiki and Web 2.0 Application Platform. [WWW]. [Cited 07.01.2011]. Available at: `http://twiki.org/`.

[3] Peter Denning, Jim Horning, David Parnas, and Lauren Weinstein. Wikipedia risks. *Commun. ACM*, 48:152–152, December 2005.

[4] Mayer R. C., Davis J. H. & Schoorman D. F. 1995. An Integrative Model of Organizational Trust. *Journal of The Academy of Management Review*, 20, 3, pp. 709-734.

[5] S. Ruohomaa and L. Kutvonen. Trust management survey. *Trust Management*, pages 77–92, 2005.

[6] A. Josang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, 2007.

[7] F.R. Farmer and B. Glass. *Building web reputation systems*. Yahoo Press, 2010.

[8] A. Jøsang, C. Keser, and T. Dimitrakos. Can we manage trust? *Trust Management*, pages 93–107, 2005.

[9] Kannisto J. 2009. Maineenhallintajärjestelmän rakentaminen yhteisöllisen opetusportaalin käyttöön. Bachelor's thesis, Department of Communications Engineering, Tampere University of Technology, Tampere, Finland. [In Finnish]. Available at: `http://jop.cs.tut.fi/twiki/pub/TLTT/KandinTyot/joona.kannisto.pdf`.

[10] GNU General Public License. 1991. [WWW]. [Cited 05.04.2011]. Available at: `http://www.gnu.org/licenses/gpl-2.0.html`.

[11] L. Rasmusson and S. Jansson. Simulated social control for secure Internet commerce. In *Proceedings of the 1996 workshop on New security paradigms*, pages 18–25. ACM, 1996.

[12] R.L. Wash. *Motivating Contributions for Home Computer Security*. PhD thesis, The University of Michigan, 2009.

[13] J. Boyd. In community we trust: online security communication at eBay. *Journal of Computer-Mediated Communication*, 7(3), 2002.

[14] Eric J. Friedman and Paul Resnick. The social cost of cheap pseudonyms. *Journal of Economics and Management Strategy*, 10:173–199, 1998.

[15] Roger Dingledine, Nick Mathewson, and Paul Syverson. Reputation in privacy enhancing technologies. In *Proceedings of the 12th annual conference on Computers, freedom and privacy*, CFP '02, pages 1–6, New York, NY, USA, 2002. ACM.

[16] Editorial oversight and control in Wikipedia. 2011. [WWW]. [Cited 9.3.2011] Available at: `http://en.wikipedia.org/w/index.php?title=Wikipedia:Editorial_oversight_and_control&oldid=415120243`.

[17] J.A. Koskinen and M. Helenius. 2010. Tietoturva web. [WWW]. [Cited 22.06.2011]. [In Finnish]. Available at: `http://jop.cs.tut.fi/twiki/bin/view/Tietoturva/WebHome?rev=24`.

[18] Sivula J., Ylä-onnenvuori R. 2008. TWiki yhteisöllisen opetusportaalin alustana - valinta, asennus ja toiminnan varmistaminen. Bachelor's thesis, Department of Communications Engineering, Tampere University of Technology, Tampere, Finland. [In Finnish]. Available at: `https://jop.cs.tut.fi/twiki/pub/TLTT/KandinTyot/sivula--yla-onnenvuori.pdf`.

[19] L. Brown and P. Thoeny 2008. Twiki Applications. [WWW]. [Cited: 6.8.2011]. Available at: `http://twiki.org/cgi-bin/view/Codev/TWikiApplication?rev=63`.

[20] Thoeny P., et al. TWiki Plugins. [WWW]. [Cited 24.1.2011]. [Modified 31.5.2010]. Available at: `http://twiki.org/cgi-bin/view/TWiki/TWikiPlugins`.

[21] About Perl. [Cited 24.1.2011] Available at: `http://www.perl.org/about.html`.

[22] M. Burke, C. Marlow, and T. Lento. Feed me: motivating newcomer contribution in social network sites. In *Proceedings of the 27th international conference on Human factors in computing systems*, pages 945–954. ACM, 2009.

[23] D. Ariely. *Predictably irrational*. Harper Collins, 2008.

[24] E. Carrara and G. Hogben. Reputation-based systems: a security analysis. ENISA Position Paper, 2007.

[25] Jerome H. Saltzer and Michael D. Schroeder. The protection of information in computer systems, 1975.

[26] P. Gutmann, "Security Usability Fundamentals," [WWW.] [Cited. 14.2.2011] Available at: `http://www.cs.auckland.ac.nz~pgut001/pubs/usability.pdf`.

[27] Lorrie Cranor and Simson Garfinkel. *Security and Usability.* O'Reilly Media, Inc., 2005.

[28] Alma Whitten and J. D. Tygar. Why johnny can't encrypt. In *Proceedings of the 8th USENIX Security Symposium*, 1999.

[29] W. Keith Edwards, Erika Shehan Poole, and Jennifer Stoll. Security automation considered harmful? In *Proceedings of the 2007 Workshop on New Security Paradigms*, NSPW '07, pages 33–42, New York, NY, USA, 2008. ACM.

[30] L.F. Cranor. A framework for reasoning about the human in the loop. In *Proceedings of the 1st Conference on Usability, Psychology, and Security*, pages 1–15. USENIX Association, 2008.

[31] Dingledine R., Freedman M.J., Molnar D., Parkes D. and Syverson P. 2004. Reputation. [WWW]. [Cited 24.1.2011]. Available at: `http://freehaven.net/~arma/jean.html`.

[32] Slashdot FAQ. 2011. [WWW]. [Cited 22.2.2011]. Available at: `http://slashdot.org/faq`.

[33] Ross A. Malaga. Web-based reputation management systems: Problems and suggested solutions. *Electronic Commerce Research*, 1:403–417, 2001. 10.1023/A:1011557319152.

[34] Zacharia G. 2000. Trust management through reputation mechanisms. *Applied Artificial Intelligence* 14, pp. 881–907.

[35] A. Chavez and P. Maes. Kasbah: An agent marketplace for buying and selling goods. In *Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology*, volume 434, 1996.

[36] B. Thomas Adler, Krishnendu Chatterjee, Luca de Alfaro, Marco Faella, Ian Pye, and Vishwanath Raman. Assigning trust to wikipedia content. In *Proceedings of the 4th International Symposium on Wikis*, WikiSym '08, pages 26:1–26:12, New York, NY, USA, 2008. ACM.

[37] A. Sharma, L. Davis, M. Pantages, E. Möller, H. Fung, B. Harris and G. Paumier 2010. Article Feedback FAQ. [WWW]. [Cited 26.07.2011]. Available at: `http://www.mediawiki.org/w/index.php?title=Article_feedback/FAQ&oldid=350428`.

[38] Roe S., et al. 2011. VotePlugin. [WWW]. [Cited: 13.4.2011]. Available at: `http://twiki.org/cgi-bin/view/Plugins/PeerPlugin?rev=13`.

[39] Daum M., et al. 2011. VotePlugin. [WWW]. [Cited: 13.4.2011]. Available at: `http://twiki.org/cgi-bin/view/Plugins/VotePlugin?rev=44`.

[40] Thoeny P., et al. 2011. TagMePlugin. [WWW]. [Cited: 13.4.2011]. Available at: `http://twiki.org/cgi-bin/view/Plugins/TagMePlugin?rev=49`.

[41] L. Mui, M. Mohtashemi, and A. Halberstadt. A computational model of trust and reputation. In *System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on*, pages 2431–2439. IEEE, 2002.

[42] D. Weller, et al. 2011. CommentPlugin. [WWW]. [Cited: 14.4.2011]. Available at: `http://twiki.org/cgi-bin/view/Plugins/CommentPlugin?rev=95`.

[43] C. Currie, S. Dowideit and PeterThoeny 2008. Twiki Applications. [WWW]. [Cited: 21.11.2011]. Available at: `http://twiki.org/cgi-bin/view/Codev/SecuringYourTWiki?rev=16`.

[44] Lintula. 2011. [WWW] [Cited: 21.2.2011] Available at: `http://www.cs.tut.fi/lintula/index.shtml.en`.

[45] The Apache Software Foundation. 2011. Apache HTTP Server Project. [WWW]. [Cited 4.4.2011]. Available at: `http://httpd.apache.org/ABOUT_APACHE.html`.

[46] Comprehensive Perl Archive Network. [WWW]. [Cited 24.1.2011] Available at: `http://www.perl.org/cpan.html`.

[47] Seppänen V. 2009. Verkkoportaalin suorituskyvyn parantaminen. Bachelor's thesis, Department of Communications Engineering, Tampere University of Technology, Tampere, Finland. [In Finnish]. Available at: `http://jop.cs.tut.fi/twiki/pub/TLTT/KandinTyot/ville.seppanen.pdf`.

[48] Apache Foundation. 2011. Welcome to the mod_perl world. [WWW]. [Cited 24.1.2011]. [Modified 24.1.2011] Available at: `http://perl.apache.org/`.

[49] Thoeny P. et al. Using Subversion to track the latest TWiki developments `http://twiki.org/cgi-bin/view/Codev/SubversionReadme`.

[50] Clemens A. et al. 2011. About Foswiki. [WWW]. [Cited 05.04.2011] Available at: `http://foswiki.org/About/WebHome?rev=38`.

[51] Ulrich A. Development of Foswiki and TWiki - get the facts [WWW]. [Cited 25.1.2011] `http://blog.wikiring.com/Blog/BlogEntry36`.

[52] Sarathy G., et al. MLDBM - store multi-level Perl hash structure in single level tied hash. [WWW]. [Cited 24.1.2011]. [Modified 8.3.2010]. Available at: `http://search.cpan.org/~chorny/MLDBM-2.04/lib/MLDBM.pm`.

[53] M.A. Olson, K. Bostic, and M. Seltzer. Berkeley db. In *Proceedings of the FREENIX Track: 1999 USENIX Annual Technical Conference*, pages 183–192, 1999.

[54] Thoeny P., et al. 2011. TWiki Installation Guide. [WWW]. [Cited: 21.3.2011]. Available at: `http://twiki.org/cgi-bin/view/TWiki/TWikiInstallationGuide?rev=282`.

[55] Thoeny P., et al. 2011. Setting Preferences Variables. [WWW]. [Cited 07.01.2011]. [Modified 26.5.2010]. Available at: `http://twiki.org/cgi-bin/view/TWiki/TWikiVariables#Setting_Preferences_Variables`.

[56] C. Currie , et al. 2011. RatingContrib. [WWW]. [Cited: 13.4.2011]. Available at: `http://twiki.org/cgi-bin/view/Plugins/RatingContrib?rev=12`.

[57] Resnick P. and Zeckhauser R. 2002. Trust among strangers in internet transactions: empirical analysis of eBay's reputation system. *Advances in Applied Microeconomics*, Vol 11, pp 127 - 157.

[58] T. Berners-Lee, R. Fielding, and L. Masinter. Uniform resource identifier (URI): generic syntax. RFC 3986, Internet Engineering Task Force, January 2005.

[59] D. Ariely. *The upside of irrationality: the unexpected benefits of defying logic at work and at home.* HarperCollins, 2010.

[60] F.N. Egger. Trust me, i'm an online vendor: towards a model of trust for e-commerce system design. In *CHI'00 extended abstracts on Human factors in computing systems*, pages 101–102. ACM, 2000.