

7-9-2009

# Automating the moderation process in GEO using trust metrics

Harihar Shankar

Follow this and additional works at: [https://digitalrepository.unm.edu/ece\\_etds](https://digitalrepository.unm.edu/ece_etds)

---

## Recommended Citation

Shankar, Harihar. "Automating the moderation process in GEO using trust metrics." (2009). [https://digitalrepository.unm.edu/ece\\_etds/231](https://digitalrepository.unm.edu/ece_etds/231)

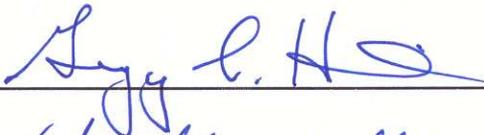
This Thesis is brought to you for free and open access by the Engineering ETDs at UNM Digital Repository. It has been accepted for inclusion in Electrical and Computer Engineering ETDs by an authorized administrator of UNM Digital Repository. For more information, please contact [disc@unm.edu](mailto:disc@unm.edu).

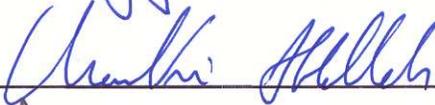
HARIHAR SHANKAR  
Candidate

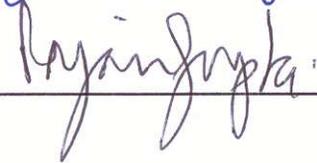
ELECTRICAL AND COMPUTER ENGINEERING  
Department

This thesis is approved, and it is acceptable in quality and form for publication:

*Approved by the Thesis Committee:*

, Chairperson





\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

# **Automating the Moderation Process in GEO using Trust Metrics**

by

**Harihar Shankar**

B.E., Anna University

THESIS

Submitted in Partial Fulfillment of the  
Requirements for the Degree of

Master of Science  
Computer Engineering

The University of New Mexico

Albuquerque, New Mexico

May, 2009

©2009, Harihar Shankar

# Dedication

*To my parents and my sisters.*

# Acknowledgments

This thesis owes its existence to the help, support, and inspiration of many people. First and foremost, I would like to thank my advisor Dr. Gregory Heileman for his guidance. I would like to express my sincere appreciation and gratitude to Dr. Rajan Gupta for his patient guidance, encouragement and excellent advice throughout this work. I am grateful to Dr. Chaouki Abdallah for agreeing to be on my committee and reviewing this work. I also thank my family and friends for their encouragement and support, especially Vasudha and Sriram, without whom I wouldn't be where I am today.

# **Automating the Moderation Process in GEO using Trust Metrics**

by

**Harihar Shankar**

## **ABSTRACT OF THESIS**

Submitted in Partial Fulfillment of the  
Requirements for the Degree of

Master of Science  
Computer Engineering

The University of New Mexico

Albuquerque, New Mexico

May, 2009

# **Automating the Moderation Process in GEO using Trust Metrics**

by

**Harihar Shankar**

B.E., Anna University

M.S., Computer Engineering, University of New Mexico, 2009

## **Abstract**

Distributed knowledge bases offer novel and fascinating ways to build, integrate and process knowledge. In these distributed knowledge bases, many users collaborate and contribute content. Verification and validation of the user's contributions become imperative for the success of these knowledge bases, especially if it is a scientific knowledge base. However, manually verifying and moderating the contributions become a bottleneck for an up-to-date system. In this thesis, we propose an algorithm to automate the moderation process and we implement the algorithm in an open scientific knowledge base application called Global Energy Observatory (GEO). Using Trust Metrics, user's contributions can be automatically accepted without waiting for the moderators to verify and validate them. This provides users with a smooth, hassle free experience when they contribute data. We also provide empirical analysis to substantiate our algorithm. An open information exchange architecture, that makes use of Semantic Web formats is also presented.

# Contents

<b>List of Figures</b>	<b>x</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Background</b>	<b>4</b>
<b>3 Managing Trust in GEO</b>	<b>9</b>
3.1 Trust Metrics . . . . .	9
3.1.1 Classification of Trust Metrics . . . . .	10
3.2 Global Energy Observatory . . . . .	11
3.3 Trust Metrics in GEO . . . . .	14
3.3.1 Trust Metrics for Contributors . . . . .	14
3.3.2 Implementation . . . . .	17
3.3.3 Trust Metrics to Rate the Accuracy of Data . . . . .	20
3.3.4 Implementation . . . . .	21

*Contents*

<b>4 Experiment and Results</b>	<b>23</b>
4.1 Experiment Setup . . . . .	24
4.2 Results . . . . .	26
<b>5 Conclusion and Future Work</b>	<b>31</b>
<b>References</b>	<b>33</b>

# List of Figures

3.1	Classification of Trust Metrics. . . . .	10
3.2	Example of the trust propagation algorithm. . . . .	16
3.3	Example of a triple represented as an RDF graph. . . . .	18
3.4	A simple representation of the rating information in XML. . . . .	19
3.5	A graphical illustration of the trust architecture. . . . .	20
4.1	The mean accuracy rating of 1000 records for the moderator rating of 1.0. The x-axis is the rating for editors in percentage and the y-axis is the accuracy rating for records in percentage. . . . .	28
4.2	The mean accuracy rating of 1000 records for moderator rating of 0.95. The x-axis is the rating for editors in percentage and the y-axis is the accuracy rating for records in percentage. . . . .	29
4.3	The accuracy rating of records for moderator rating of 0.90. The x-axis is the rating for editors in percentage and the y-axis is the accuracy rating for records in percentage. . . . .	30

# Chapter 1

## Introduction

Distributed knowledge bases such as wikis are becoming increasingly more attractive because of the way they offer fascinating ways to build, integrate and process knowledge. In these distributed knowledge bases, many users collaborate and contribute content. These distributed knowledge bases, however, pose a critical challenge: to verify and moderate the content users contribute. In a wiki, where any user can contribute content, it is often the case that not every user will know every other user, and hence it also becomes difficult to trust the data an unknown user contributes. It is imperative in a wiki to provide a formal way for users to declare their confidence in other users. At the very least, it provides an additional piece of information that may be useful when coming across another user for the first time. This is achieved by Trust Metrics. Trust metrics try to automate the word of mouth reputation. Trust metric computes a global trust-worthiness value for each user, based on the information of how much trust other users have on this user.

Global Energy Observatory (GEO) [13] is a moderated distributed knowledge base, which will act as an observatory to monitor and reference global energy systems and their emissions. The scientific nature of data in GEO requires manually verifying and validating the data that users contribute. This process of manually moderating the contributions

## *Chapter 1. Introduction*

has many drawbacks and is a bottleneck for the smooth operation of a system. In this thesis, we apply trust-metrics to automate the moderation process in a scientific distributed knowledge base.

Trust metrics have been used to automate the moderation process for news websites and other general community forums. Slashdot [14] is one such technology related news website. Slashdot has automated the moderation process, whereby, every comment posted has a starting score which can be incremented or decremented by semi-randomly chosen moderators. Trust metrics are also widely used in recommender systems: a specific type of information filtering technique, that attempts to present information items that are likely of interest to the user. Moleskiing.it [6] is one such recommender system that uses trust metrics to provide recommendations for users in the ski mountaineering domain. However, trust metrics have not been used to automate the moderation process in a scientific distributed knowledge base.

In this thesis, we present an algorithm which computes a trust-worthiness value of a user based on the trust ratings other users provide for this user. Trust values are provided to a user by other users based on the contribution the user makes. When the trust value of a user reaches more than a threshold value, then further contributions of that user does not wait for moderation and is directly accepted.

In addition to automating the moderation process, we also implement trust metrics for users to rate the accuracy of the data in the records of GEO. A record in GEO is a webpage displaying information about every power system around the world. A record may contain contributions from more than one user and may contain many fields of data. This trust metric helps us to determine how accurate the records in GEO are.

We have implemented both these trust metrics in open, decentralized Semantic Web [1] formats, enabling us to potentially aggregate or export the trust information of a user from other systems and not restrict it to GEO.

## *Chapter 1. Introduction*

We also performed empirical analysis to understand the relationship between these two trust metrics: the trust metrics on contributors and the trust metrics on the accuracy of the data the users have contributed. We determined the minimum trust rating the users in GEO should work on so that the eventual accuracy of data is over 90%.

The use of trust metrics in GEO not only enables us to overcome the drawbacks of manual moderation, but it also provides a platform for users to trust other users in the system. The use of semantic web format, open fascinating ways to use the trust data on users in GEO and export it to make collaborations with many other similar systems.

The following chapter gives an overview of distributed knowledge bases and discusses all the related work that has been done in this field. Chapter 3 explains trust metric in detail, gives a broad overview of the GEO application and then explains the algorithm for GEO in detail including details about its implementation. Description of the empirical analysis and its results are given in Chapter 4.

# Chapter 2

## Background

Web-based distributed knowledge bases or wikis are changing the way in which data and knowledge are integrated and processed. A wiki is a collection of web-pages designed to enable anyone who accesses it to contribute or modify its content. They are often used to create collaborative websites and to power community websites. Wikipedia [15] is one such popular collaborative website. Wikis are also used widely in companies to provide intranet and knowledge management systems. Wikis can be classified into three major categories based on their usage. Educational wikis serve the purpose of educating people through an open and collaborative publication format. Social wikis allow people to discuss and publish information about a social subject of interest. Corporate wikis are used by groups or teams that work in collaborative environments for storing online documentations or minutes of meeting.

The most important aspect of a wiki is the ability to change and modify its content immediately, any page in the wiki can be modified easily using a web browser using the edit feature provided by the wiki. Changes are then directly presented after saving the modifications. Representing knowledge and information in a wiki has the following benefits. Wikis are simple and the interface is intuitive. Contributing to a wiki does not require

## *Chapter 2. Background*

any installation and only requires a standard web browser. It can combine both textual descriptions and multimedia information to represent knowledge in a wiki page. Standard wiki tools like version control, statistics and query engines lets multiple developers work on a complex system.

A wiki can also be classified into a normal wiki or a moderated wiki based on the rights a user has, and the behavior of the wiki when a contribution is made. The differences between these two are as follows. In a normal wiki, any user can contribute or modify the wiki and the changes are immediately presented. However, in a moderated wiki, there are two types of users: the regular user and moderators. Moderators are a group of users chosen based on their expertise to act as delegates to enforce the community rules of the wiki. The contribution of users are not directly presented and they wait for the approval of the moderators.

In recent times, social semantic systems such as semantic wikis are gaining in popularity. Semantic wikis offer great ways to build intelligent systems in a collaborative way. Here, the systems are not used to build a single knowledge base but a federation of knowledge bases that are able to work together on complex problems. Semantic knowledge wikis extend traditional wikis by the representation of explicit problem-solving knowledge. DBpedia [16] is one such semantic knowledge wiki.

GEO is an effort to build a moderated wiki, that will act as an observatory to monitor and reference global energy systems and their emissions. GEO provides the ability for users to add, edit and analyze information about any power system in the world including information regarding power generation, emission, ownership information, etc. This ability to incorporate different kinds of data and provide tools to perform queries and analyses on up-to-date data, makes it a unique scientific wiki. The scientific nature of the data mandates constant verification and moderation of user contributions. In GEO, Editors and Moderators (Section 3.2) are manually chosen from the pool of users based on their expertise and their contributions to the system. These editors and moderators have the privilege

## *Chapter 2. Background*

to verify and validate the contributions of other users.

To manually verify and moderate each contribution has the following disadvantages. These include the fact that the entire success of the system is based on the performance of the moderators. In a real-time system like GEO, irregular moderation may lead to inconsistent information. Moderators may be biased towards certain contributors and this may again lead to inconsistency in data and may even act as a deterrent for users from using GEO. Tracking the contribution and the expertise of a user manually to be chosen as a moderator is tedious and time-consuming.

Hence, an automated moderation process with minimum human intervention is the key determinant to the success of this system. We implemented an algorithm using trust metrics for this purpose. A trust metric is a measure of how a member of a group is trusted by the other members. In GEO, any user can rate a contributor based on his contributions. The rating value is the “trust” a user has about another user. When a user’s trust value reaches a threshold, then further contributions from this user will be automatically approved and will be presented to other users immediately. This algorithm overcomes the drawbacks of manual moderation and it chooses worthy users automatically, based on their contributions.

A general form of a trust management system was developed and implemented in the PolicyMaker system [11]. Prior to this “trust management” was not identified as a problem domain and cryptographic techniques were used. More precisely, trust-based systems were based on public key certificates, where a trusted third party provides a signed message called “certificate” to identify the associated public key. PolicyMaker provides a common language for policies, credentials and relationships. Applications of PolicyMaker tends to focus on the language of assertions rather than computing trust over a graph.

Advogato is a popular local trust metric (Section 3.1) algorithm. It is a network flow based algorithm and is used to decide which members are trustworthy and competent open-

## *Chapter 2. Background*

source developers. Advogato calculates trust using a concept called group trust metric. This trust metric takes a directed graph as an input, where the users form the nodes and their trust on another user forms the edge between the nodes. The group trust metric calculates the trust value for all the nodes in the graph at once, rather than calculating independently the trust value for each node. In GEO, however, we calculate the trust value of a user only from the point of view of Editors/Moderators (Chapter 3). The calculation of trust values for all the other nodes is not necessary and is also computationally expensive.

Recommender Systems (RS) suggests to users items they may like, based on users' previous opinions. RS is very similar to the moderation system in GEO, except RS's deal with users' opinion and comments rather than scientific data. Moleskiing.it [7] is one such RS that uses trust metrics in Ski Mountaineering domain. Moleskiing.it is a web application which allows users to share their experiences on a ski route and provides personalized opinions to users seeking information on a route, based on the users they trust. However, this system cannot be directly applied to the moderation scenario in GEO as it is a knowledge base rather than a forum for comments.

The moderation process in GEO is also similar to the peer-review process. Peer-review process is the standard for validating the written results of researchers within the scientific community by identifying competent referees to review, accept or reject the manuscripts. Rodriguez and Bollen [12] designed an algorithm to determine peer-reviewers. The proposed algorithm provides a context-specific weight for every individual represented in the co-authorship network, where the context is the document required for review. The algorithm identifies appropriate referees for a manuscript based on the premise that a manuscript's subject domain can be represented by the authors of its references. Starting from those authors, the algorithm identifies related authors in a co-authorship network who may be potential referees for the submitted manuscript. However, in GEO, we may encounter different levels of users from an expert in the topic to the casual user and all users are treated equally. Hence, a network of trust for users is critical.

## *Chapter 2. Background*

The following chapter gives an overview and classification of trust management, followed by the overview of GEO and the detailed explanation of the trust metric algorithm with its implementation.

# Chapter 3

## Managing Trust in GEO

This chapter discusses trust metrics and gives a broad overview of GEO. It also explains in detail the two trust metrics that were implemented in GEO including the details of its implementation.

### 3.1 Trust Metrics

In this section, we present a simple form of trust metric. This forms the base for the trust metric that is later discussed in this thesis.

Trust metrics have three inputs: a directed graph, a root node and the target node. We can determine how much the root node trusts the target node.

In a directed graph  $g$ , an edge from seed  $s$  to the target  $t$  implies that  $s$  believes that  $t$  is trustworthy. The simplest implementation of trust metrics evaluates whether  $t$  is reachable from  $s$ .

This implementation of trust metric is very weak and prone to attacks. If an attacker could generate an edge reachable from the seed to a node under his control, then trust

could be compromised. Hence, we have focussed on making a stronger trust metric that resist attacks.

### 3.1.1 Classification of Trust Metrics

There are two major classification of trust metrics: global and local [3]. Local trust metrics calculates trust with a personal bias: trust is computed from a particular user's position in the trust graph. The Advogato trust metric [2] is a local trust metric. On the other hand, in global trust metrics, trust values are not personalized. Global trust ranks are assigned based on complete information from the trust graph. Nodes that has better ranked nodes pointing at them, has higher ratings. The most popular global trust metric is the PageRank algorithm [4].

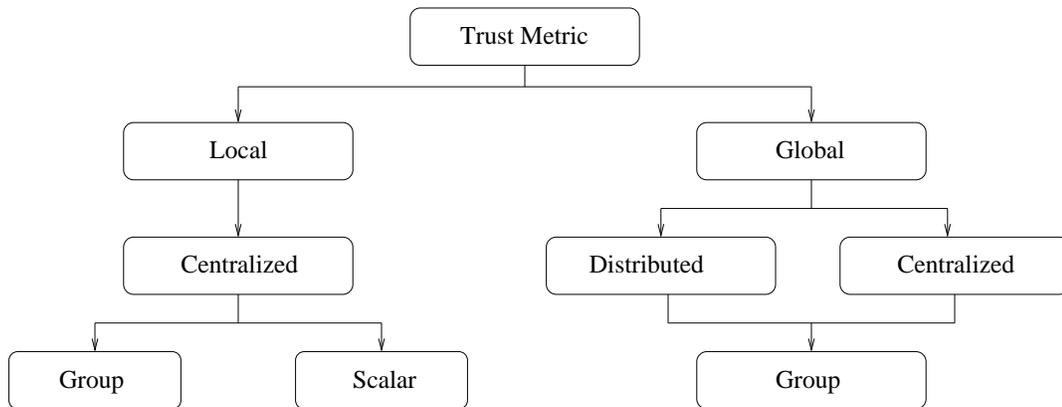


Figure 3.1: Classification of Trust Metrics.

The global and local trust metrics can further be categorized into centralized and distributed trust metrics. All trust computations are performed in a single machine in a centralized approach and hence this computer will be granted full access to the trust information. In a distributed setting, such as [5], the computation of trust is equally distributed among all the nodes in the trust graph.

These can be further classified into scalar and group trust metrics. Scalar metrics analyze trust assertions independently, while group trust metrics evaluate the trust value for all nodes at once [2]. The classification [3] is shown in Figure 3.1.

Another important feature of local trust metric is the fact that it is attack-resistant [2, 6]. Users who are considered malicious from another user's perspective, do not influence others and are excluded from trust propagation. This property is especially important for a system like GEO where a malicious user should not be allowed to influence other users to climb up the hierarchy. A more detailed explanation of trust metrics and attack resistance can be found in [6].

## **3.2 Global Energy Observatory**

Energy, development and its impact on environment and climate is, perhaps, the most urgent and multi-dimensional challenge that is faced today. Until technological innovations can provide environmentally sustainable and reasonably priced energy to the global population, it is imperative that tools are developed to help quantify and understand the technological, sociological, economic and political aspects of the challenge and motivate investment in energy efficiency and lifestyle changes that would reduce use of fossil fuels and emissions of greenhouse gases.

Global Energy Observatory (GEO) is an effort to build, with public participation, a web-based real-time observatory of geo-spatially referenced global energy systems and their emissions. GEO will also be used to educate people and motivate them to change their lifestyles in order to promote reduced fossil fuels, and encourage the development of alternate energy technologies. The aim is to make GEO a “one-stop shopping” for adding, correcting, accessing energy and emissions data, and include helpful tools for data analysis.

### *Chapter 3. Managing Trust in GEO*

Lots of information on energy systems are available in digital form but they are highly fragmented and the quality and detail are not uniform. Data are available in many different formats including Excel, PDF, XML, and HTML making it difficult to integrate, analyze and visualize data. GEO provides a comprehensive framework for collecting, completing, managing and validating this data.

The current version of GEO has been implemented using Linux-Apache-MySQL-PHP (LAMP) infrastructure. GEO can be classified into 5 major sections:

- **Large Power Plants:** Consists of commercial scale power plants including Coal, Gas, Geothermal, Hydro, Nuclear, Oil, Solar PV, Solar Thermal, Waste and Wind.
- **Fuels and Resources:** Includes Gas and Oil fields, Coal and Uranium mines, Crude Oil refineries, Solar and Wind potential.
- **Transmission of energy:** Includes Gas and Oil pipelines, Coal and Oil ports, Rail and Road links, shipping lanes and Electric Transmission grid.
- **Distributed Generation:** Keeps track of Solar, Wind and Geothermal systems for individual homes and institutions.
- **Carbon Footprint** of individual end-users and their understanding of energy and its impact on environment.

Large amount of open data is already available in the internet for the first 3 sections listed above. Significant amount of this open data has been harvested and incorporated into GEO. The long-term goal of this project is to involve users of GEO to contribute and maintain the data in this database. The success of GEO depends on the quality of the data and its usefulness. The data in GEO is heterogeneous: data from disparate sources and data formats. This makes validation, verification and moderation essential for GEO. However, the key determinant of success to a project of this magnitude and impact, will be to make it self-sustaining with minimal supervision.

### *Chapter 3. Managing Trust in GEO*

The information (say for a power plant in New Mexico) that is available to every user in GEO is the most recently moderated version. The contribution of users to a record (information about a power plant, for example) will not become the final version and be available for other users to view immediately. A moderator should review and approve the content. A moderator is an expert in a particular field and a particular country. He/she has powers to accept or reject contributions from users in his/her domain of expertise. This system is inefficient, especially in a large project with hundreds of users. This gives us a unique problem and hence we have chosen to modify and enhance existing trust-metrics to automate the moderation process in GEO.

The users in GEO are classified into 5 major categories.

- **Casual users:** These are casual visitors who visit GEO from time to time.
- **Contributors:** These are more regular users who also contribute data to GEO.
- **Editors:** Expert users who review contributions from users, create new consolidated versions from many contributions and provide suggestions to moderators.
- **Moderators:** Expert users who has the authority to accept/reject changes based on the editor's suggestions or on their own discretion. Moderators can accept/reject changes only in their field of expertise (Coal power plants, for example) and in their country of expertise.
- **System Administrators:** Users who have complete control over the system.

Editors and Moderators can rate the user based on their contribution. This rating will depend both on the quality and quantity of the contribution. In this way, a user builds trust in GEO. Depending on the trust rating of the user, future contributions will be automatically accepted into the database without waiting for the approval of the moderator. Note that the user do not get promoted to a moderator. Only the contributions are automatically

accepted without moderation. Every user is allowed to rate other contributors. However, a user is not expected to receive ratings from a large number of other users. Hence, the ratings of other users are only used as a guide by the moderator when rating a particular user and it is not used in the actual computation of the rating.

The trust statement of a user is the determining factor of this system. In order to facilitate ease of access to this information, we have implemented the trust-metrics in open, decentralized Semantic Web [1] formats. This enables us to potentially aggregate or export the trust information of a user from other systems and not restrict it to GEO.

### **3.3 Trust Metrics in GEO**

We have used trust metrics for two different purposes in GEO: Trust Metrics for contributors to automate the moderation process and trust metrics to rate the accuracy of data. This section explains both these trust metrics.

#### **3.3.1 Trust Metrics for Contributors**

A preliminary trust metric for automation of the moderation system is provided in this section, with details of its implementation. This algorithm will be modified, improved and compared with other trust metrics as and when more people start using it and we have real usage data.

Trust statements are weighted and measured from distrust to trust in the real interval  $[0,1]$ . There are two aspects to this algorithm. First, there is a straight forward rating system for moderators and editors providing trust ratings on contributors. Second, there is a recommendation system for moderators and editors based on the ratings provided by other users on contributors. This follows trust propagation [6]: allows us to infer a

### Chapter 3. Managing Trust in GEO

personalized trust score for unknown users.

The first part of the algorithm works as follows. The moderators and/or editors are allowed to provide ratings on users depending on their contribution. In this process of rating, they are also expected to provide information on the country and domain that the user is being rated upon. For example, a user  $A$  may have a rating of  $0.8$  in *Coal Power Plants* in *USA*. Note that, the same user can have different ratings for different domains (like Coal, Nuclear, etc.) and for different countries (like USA, Canada, etc.). If a user has a rating of more than  $0.9$ , the next time he submits contribution in that country and domain of expertise, the contribution gets immediately accepted into the system with a probability of his rating score. The rating score is calculated by a simple walk of the trust graph and computing the average of the ratings provided by the moderators and editors. Since the height of the graph from the user is just one, the computation is less and there are no possibilities of cycles in the graph. Moreover, only the ratings of the editors and moderators are considered for the actual computation of the trust metric. Hence, attacks by malicious users are eliminated.

The recommendation system for the moderators and editors is a modified version of the trust metric, MoleTrust [7] and it works in two steps. First, the cycles in the graph are removed. An example of a cycle is when user  $A$  trusts  $B$   $0.9$ ,  $B$  trusts  $C$   $0.6$  and  $C$  trusts  $A$   $0.7$ . When computing the trust, cycles create passing over these nodes many times until the value converges. This makes the computation unnecessarily complex and time consuming. Hence, a new parameter, the trust propagation height, is introduced in the algorithm. This makes sure that the graph is traversed only up to a particular height when computing the trust value. This ensures the cycles are eliminated and also decreases the computation time. The intuition behind this is that, as the height of the graph increases, the propagated trust decreases. Also, always the shortest path is used between two nodes in the graph.

At the end of the first step, we get a directed acyclic graph. The next step involves a graph walk to compute the trust prediction. The trust score of all users at distance 1 is

calculated before the trust score is calculated for the users at the distance 2. The trust score of a user at distance  $n$  only depends on the user at distance  $n-1$ , which is already computed. For a node to be considered in the trust propagation, its incoming edge should have a trust value of at least  $0.6$ . This value is considered as the minimum rating a user should have to be considered trust-worthy. The predicted trust score of a user is the average of all the trust values of the incoming edges, weighted by the trust score of the user who has requested for the trust score. This given by the formula below.

$$trust(u) = \frac{\sum_{i \in predecessors} (trust(i) * trust\_edge(i, u))}{\sum_{i \in predecessors} (trust(i))}$$

To explain this with an example, the calculation of the predicted trust score of a moderator  $A$  on an unknown user  $E$  in the Figure 3.2 is done by:

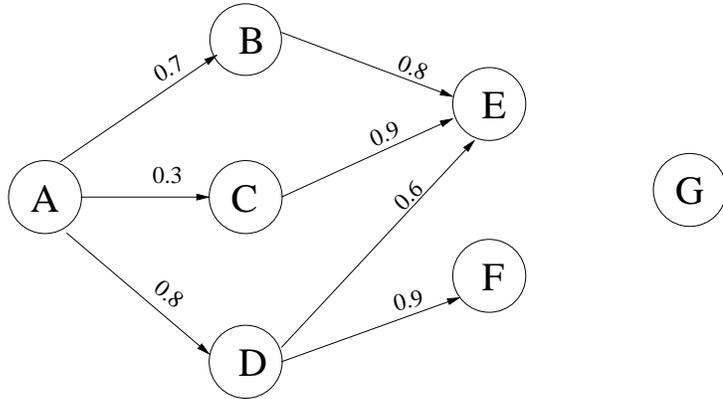


Figure 3.2: Example of the trust propagation algorithm.

$$trust(E) = \frac{0.7*0.8+0.8*0.6}{0.7+0.8} = 0.69$$

Note that the trust value of  $C$  on  $E$  is not considered since  $A$  do not trust  $C$ :  $A$ 's trust rating for  $C$  is below the threshold of  $0.6$ .

### 3.3.2 Implementation

We have implemented this trust metric using the open-source PHP package “RAP” [8]. RAP is a software package for parsing, querying, manipulating, serializing and serving the Resource Description Framework (RDF) [9] models. RDF is a framework for representing information in the Web. RDF has an abstract syntax that reflects a simple graph-based data model. This algorithm was implemented in RDF for the following reasons:

- It has a simple data model.
- It has a formal semantics and provable inference.
- It uses an extensible Universal Resource Identifier (URI)-based vocabulary.
- It uses an Extensible Mark-up Language (XML) based syntax.
- It supports the use of XML schema datatypes.
- It allows anyone to make statements about any resource.

The underlying structure of any expression in RDF is a collection of triples: consists of a subject, a predicate (also called a property that denotes a relationship) and an object. This triple is called an RDF graph. It can be represented as a “node-arc-node“ link as shown in Figure 3.3.

The nodes of the RDF graph are it’s subjects and objects and the direction of the arc always points towards the object. A node may be a URI with optional fragment identifier (URI reference), a literal or blank. The properties are URI references. A URI reference or a literal used as a node, identifies what that node represents. A URI reference used as a predicate identifies a relationship between the nodes it connects.

In GEO every user’s information is stored in RDF using Friend-Of-A-Friend (FOAF) [10]. The FOAF schema is an RDF vocabulary that a web user can use to describe

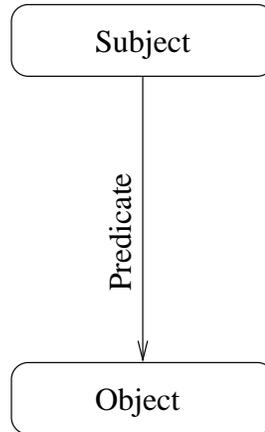


Figure 3.3: Example of a triple represented as an RDF graph.

information about himself as well as about the people he knows. People are identified in FOAF by their email-address. However, in GEO every user is required to register and login before they can provide trust ratings on other users and providing an email-address is optional. Hence, the username of a user is used as the unique field to identify users. Moreover, an additional property has been added to FOAF called `foaf:Privelege`. This property denotes the category (like moderator, editor, etc) a user belongs to, in GEO.

In addition to FOAF, GEO has it's own vocabulary called `openmodel`. This vocabulary helps define the entire rating system in GEO. `openmodel` has the following properties.

- **UserRatingID:** This property defines a class of identification numbers (ID) which uniquely identifies every rating information provided by a user for another user.
- **UserRatedBy:** This is the property that defines the `UserID` of the user providing the rating.
- **UserRating:** This is the property that defines the value of the actual rating.

### Chapter 3. Managing Trust in GEO

- **UserTimeStamp:** This is the property that defines the time and date when the rating was provided.
- **UserPlantType:** This property defines the Plant Type (like Coal, Nuclear, etc) that the user is providing a rating for.
- **UserCountry:** This property defines the country that the user is providing a rating for.

An XML snippet of a user's information and that user's rating on another user is shown in figure 3.4.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:openmodel="http://openmodel.newmexicoconsortium.org/openmodel#">
  <foaf:Person>
    <foaf:accountName>jsmith</foaf:accountName>
    <foaf:firstName>John</foaf:firstName>
    <foaf:lastName>Smith</foaf:lastName>
    <foaf:holdsAccount>http://openmodel.newmexicoconsortium.org/</foaf:holdsAccount>
    <foaf:expertise>Energy Systems</foaf:expertise>
    <foaf:highestDegree>PhD.</foaf:highestDegree>
    <foaf:city>Albany</foaf:city>
    <foaf:country>USA</foaf:city>
    <foaf:state>New York</foaf:state>
    <foaf:Privilege>Editor</foaf:Privilege>
  </foaf:person>
  <openmodel:Rating>
    <openmodel:UserRatingID>11101227</openmodel:>
    <openmodel:UserRatedBy>rjacob</openmodel:UserRatedBy>
    <openmodel:UserRating:>0.70</openmodel:UserRating>
    <openmodel:UserTimeStamp>1233298398</openmodel:UserTimeStamp>
    <openmodel:UserPlantType>Coal</openmodel:UserPlantType>
    <openmodel:UserCountry>USA</openmodel:UserCountry>
  </openmodel:Rating>
</rdf:RDF>
```

Figure 3.4: A simple representation of the rating information in XML.

The `UserRatingID` is always unique for the two users involved in the rating. In the example shown in Figure 3.4 *jsmith* will be associated with the same `UserRatingID` when *rjacob* is providing the rating. Now, this `UserRatingID` is used to capture the rating information provided by *rjacob* on *jsmith* in an RDF environment. This `UserRatingID` is significant because, we are trying to represent 4 variables: the user who is being rated, the rator, what the user is being rated upon and the value of rating, in a subject, object and predicate (triple) statement.

The figure 3.5 gives a graphical illustration of the trust architecture. An ellipse represents a URI and a square represents a literal.

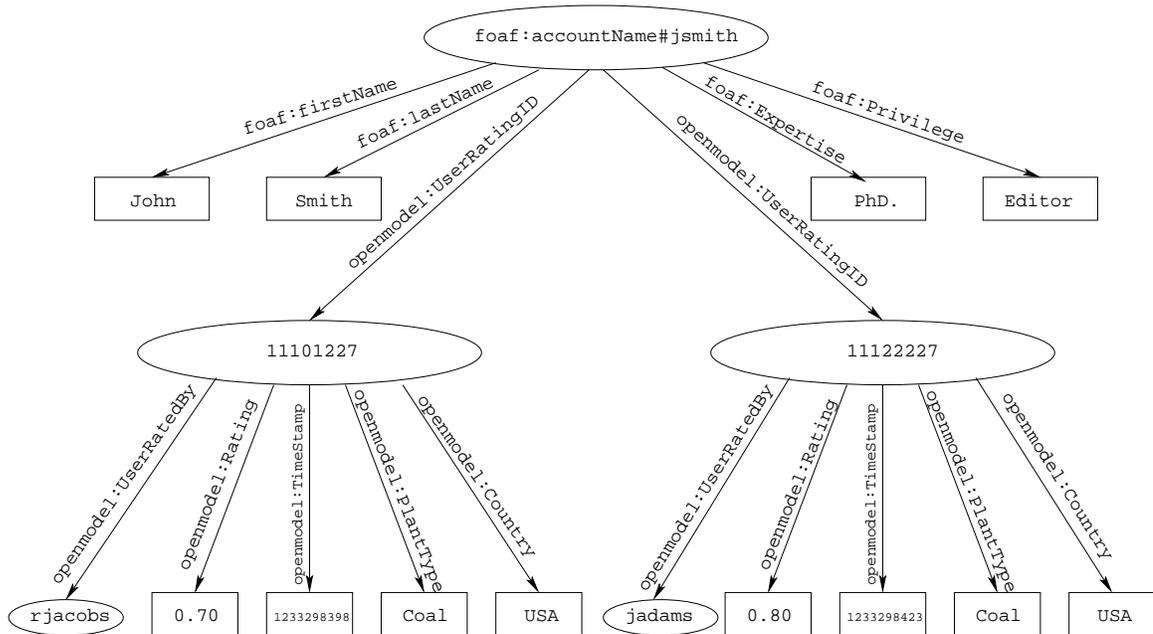


Figure 3.5: A graphical illustration of the trust architecture.

### 3.3.3 Trust Metrics to Rate the Accuracy of Data

In addition to capturing the trust on contributors, GEO also captures the trust on data or more appropriately the accuracy of data for individual records. GEO has information about energy systems (a power plant in New Mexico, for example) from all around the world and this information can be viewed, edited and retrieved by users in a web-page. Each of these pages are considered a record. Typically, the contributions of many users constitute a record. As mentioned earlier, the nature of data in these records are heterogeneous. Hence, it is necessary to keep track of the accuracy of data for every record so that users of GEO can trust the information and be motivated to further contribute and use the data.

### 3.3.4 Implementation

Users are provided with a simple interface to provide accuracy rating on every record. The rating information is stored and retrieved using RDF and PHP through RAP. The architecture of this algorithm is similar to the trust metric for users. The RDF vocabulary `openmodel` is extended to include additional properties. A brief explanation about these additional properties are listed below.

- **ParentPlantID:** This property defines the class of parent `Plant ID`: a unique identification number provided to each record in GEO.
- **hasChild:** This property defines the class of `Plant ID` that has the `ParentPlantID` as their root. The first version of every record is the parent and every subsequent contribution to the parent creates new children.
- **ratedBy:** This property is similar to `UserRatedBy`. Defines the class of users who rated a record.
- **ratingID:** This property is similar to `UserRatingID`. It defines the class of IDs which uniquely identifies the rating information by users on records.
- **PlantID:** This defines the class of IDs which identifies the record that is being rated.
- **Rating:** This property defines the rating value.
- **TimeStamp:** This property defines the time and date when the rating was provided.

In GEO, when a record is edited and submitted for moderation, a new version of the record is created. Then the first original record is called the parent  $p$  and its edited version is called its child  $c_1$ . If  $c_1$  is accepted by the moderation, then it becomes the default version of that record. Now, if any modification is done to this record, a new version of  $c_1$ ,

### *Chapter 3. Managing Trust in GEO*

$c_2$  is created. Hence, every contribution to a record generates many children  $c_1, c_2, \dots, c_n$  with its parent as  $p$ .

This rating system, hence will keep track of the accuracy rating for the parent and each of its child versions. `ParentPlantID`, `hasChild` and `PlantID` are used for this purpose.

The accuracy rating that is obtained are displayed as a graph over time and also as a histogram on every record. Currently, no trust metrics are used on these ratings. Inferences can be made using the ratings provided on the contributors and the ratings provided on the accuracy of the data in the records. However, the current implementation and settings, limits us from associating every piece of data provided, to its contributor.

# Chapter 4

## Experiment and Results

Experiments were conducted to understand the relationship between trust metrics for contributors (Section 3.3.1) and trust metrics to rate the accuracy of data (Section 3.3.3). In this chapter, the experiment and its results are discussed in detail.

In GEO, a record has many fields. For example, a typical power plant record may have hundreds of fields capturing information ranging from ownership to power production to emission details. Also, many users may have submitted data to each of these fields to constitute a record. Ideally, data in every field should be tracked to the user who has submitted that data. However, in a LAMP implementation, tracking each field in a record to its source requires a lot of overhead and is inefficient, especially when there are lots of fields. Hence, in the current implementation of GEO, we have not implemented a way to relate the rating a user has obtained for his contributions, with the rating for the accuracy of data the user has submitted. Consider a situation where a user received a rating of *0.8* for his overall contributions while the accuracy for the data he submitted for a record was only rated *0.5*. This rating of *0.5* should affect his overall rating of *0.8*. However, a record may contain contributions from more than one user and there is no efficient way to keep track of which user submitted which field in a record and how accurate each field is.

## *Chapter 4. Experiment and Results*

The aim of this experiment is to compute the minimum percentage of trust rating the users in GEO should have, so that the records maintain a certain level of accuracy rating for their data. In this experiment, the minimum percentage of accuracy for each user level (like Moderators, Editors and Contributors) is computed, so that the record they contribute to will have an accuracy rating of atleast 90%. This experiment, hence, helps us understand the relationship between the two different trust metrics that were implemented in GEO.

### **4.1 Experiment Setup**

This experiment tries to emulate the user setup in GEO as closely as possible. There are 3 levels of users in this simulation: Moderators, Editors and Contributors. There are 100 contributors, 10 editors and 1 moderator. A contributor is a user in GEO who makes regular contributions. An editor is an expert user who reviews contributions from other users, provide ratings for those users and makes a new consolidated version of all the contributions. This is really effective when many users contribute to the same record and there are many versions of the same record waiting to be moderated. The moderators accept/reject the consolidated version of the editors, in addition to accepting/rejecting the contributions from other users and provide ratings for them.

In this experiment, the moderator verifies and/or adds to the contributions of the 10 editors. Each editor consolidates and/or adds to the contributions of 10 contributors. Hence, it can be assumed that each editor reviews the work of 10 contributors and the 10 editors will review the work of all the 100 contributors. Of the 100 contributors, it is assumed that each contributor will contribute to 10 records. Hence, the 100 contributors will create 1000 records. For simplicity, it is assumed that one record will be edited by only one contributor. Ratings are generated randomly for each of the Moderators, Editors, Contributors and Records. Moreover, moderators and editors have two trust ratings each. One trust rating is a measure of how well they perform verification and validation roles and is called

## Chapter 4. Experiment and Results

`VerificationRating`. The other is a measure of how accurately they can add to the already made contribution by other users. This is called `AdditionRating`. These two ratings have been added to the experiment to make the simulation program as realistic as possible. Some editors in GEO may only verify the data submitted to them and some editors may also add/modify the data. However, the trust rating for editors and moderators in the trust algorithm explained in Section 3.3 includes the rating for both verification roles and addition roles.

The working of the simulation can be split into two major steps. First ratings are provided to each of the users using a random number generator. As in GEO, these ratings are a value between  $0$  and  $1$ . A rating of  $1$  meaning fully trustworthy. The second step is the most important step of calculating the accuracy ratings of the records based on the ratings of the users. Since every record is not assumed to have contributions from more than one user, the rating of the contributor will also imply the accuracy rating of the data of the record.

The Editor has two options: to review and verify the contribution of the contributor or to verify and add more data to the record. The program decides which action to take based on the respective ratings. If the `AdditionRating` of the editor is less than the rating of the contributor, then the program only allows the editor to verify and validate the record. In this case, the accuracy rating of the record is calculated by the average of the contributor rating and the editor's `VerificationRating`. If the `AdditionRating` of the editor is more than the rating of the contributor, then the program allows the editor to verify and add to the record. Now, the accuracy rating of the record becomes the `AdditionRating` of the editor. The intuition behind this is, if an editor adds to and modifies an already existing record, then the record is as accurate as the editor's accuracy rating.

After the record is verified by the editor, it passes on to the moderator. The moderator also has a `VerificationRating` and `AdditionRating`. The program calculates the accuracy of the record similar to that of an editor. The accuracy rating for the record

## Chapter 4. Experiment and Results

calculated here is the final accuracy rating. This program iterates to calculate the accuracy rating for 1000 records. After the accuracy for 1000 records are calculated, the mean is determined. This process is repeated with different rating values until the mean accuracy rating calculated for the 1000 records reaches atleast 90%. The mean of the ratings of moderators, editors and contributors is the average percentage of ratings the users should attain to obtain the minimum accuracy ratings of 90% for the records.

This experiment was implemented using Java.

### 4.2 Results

The goal to achieve the minimum accuracy rating of 90% for the records depend on 3 rating factors: the rating of the moderators, editors and contributors. These three variables make it difficult to express and understand the results. Therefore, for the sake of simplicity, the mean ratings of the contributors is assumed to be  $0.5$ . In other words, the contributors provide data that is on an average 50% accurate. Also, the accuracy of records is calculated assuming the moderator ratings are  $0.90$ ,  $0.95$  and  $1.0$ . In GEO, moderators are users who have a rating value of atleast  $0.9$ . Hence, assuming the rating value to be over  $0.9$  for moderators is a valid assumption.

Taking all these assumptions into account, the accuracy of data is represented as a function of the rating of editors for various moderator ratings.

Figure 4.1 plots the accuracy rating of records against mean editor rating, assuming the moderators mean rating value is  $1.0$ . As is evident from the graph, the mean record accuracy is not a linearly increasing function. This is because, the ratings for moderators, editors and contributors are randomly generated. Hence, even though the mean ratings for contributors is assumed to be 50%, it may have different variance for two data points in the graph. Also, the `VerificationRating` and `AdditionRating` also affect the

#### *Chapter 4. Experiment and Results*

mean record accuracy. The most important conclusion from the graph is that the editors should possess a minimum of approximately 80% to 81% rating provided the moderators are working at a mean rating of 100% and the contributors are working at a mean rating of 50%.

Figure 4.2 shows the accuracy rating versus editor rating for the mean moderator rating value of  $0.95$ . Note that editors should have a mean rating of at least 85% to attain overall record accuracy of 90% since the moderators are working only at 95%.

Figure 4.3 shows the same accuracy rating against editor ratings but this time for moderator ratings of  $0.90$ . This time, the mean editor ratings should at least be 88%.

To summarize, this experiment helped us to understand better the relationship between the data accuracy rating and the contributor's ratings that were discussed separately earlier. From this experiment, it can be concluded that the moderators and editors should have very high ratings for the success of GEO. Even if the moderators and editors do not contribute data themselves, they can still maintain the integrity of the records and the system by simply verifying and validating the contributions from other users.

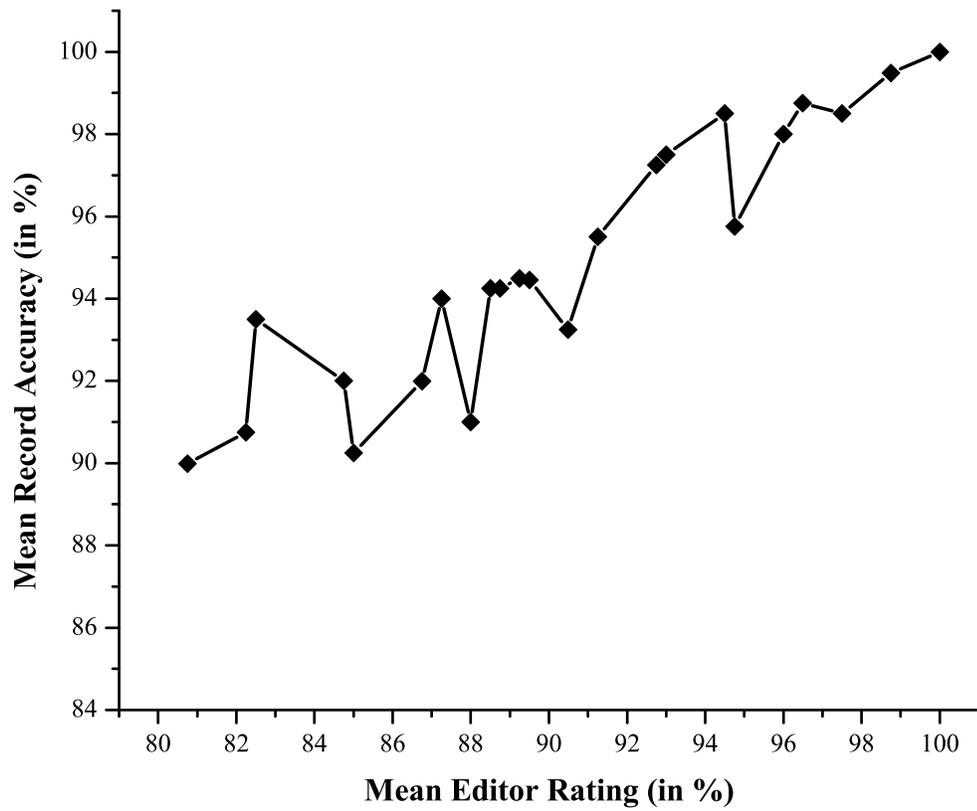


Figure 4.1: The mean accuracy rating of 1000 records for the moderator rating of 1.0. The x-axis is the rating for editors in percentage and the y-axis is the accuracy rating for records in percentage.

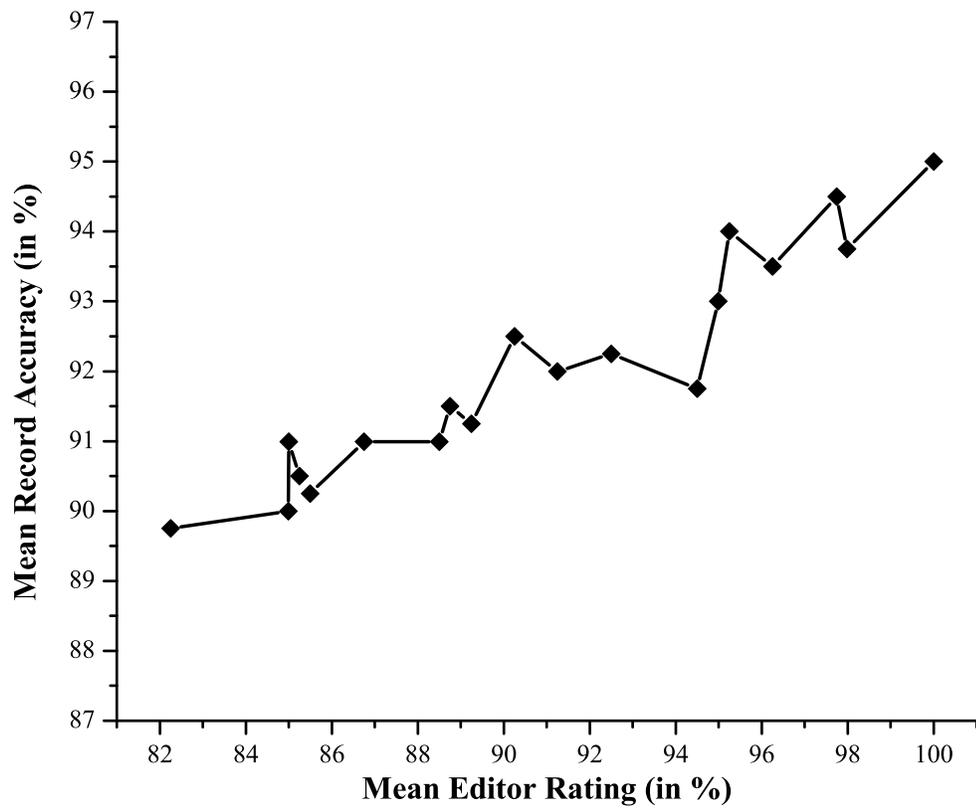


Figure 4.2: The mean accuracy rating of 1000 records for moderator rating of 0.95. The x-axis is the rating for editors in percentage and the y-axis is the accuracy rating for records in percentage.

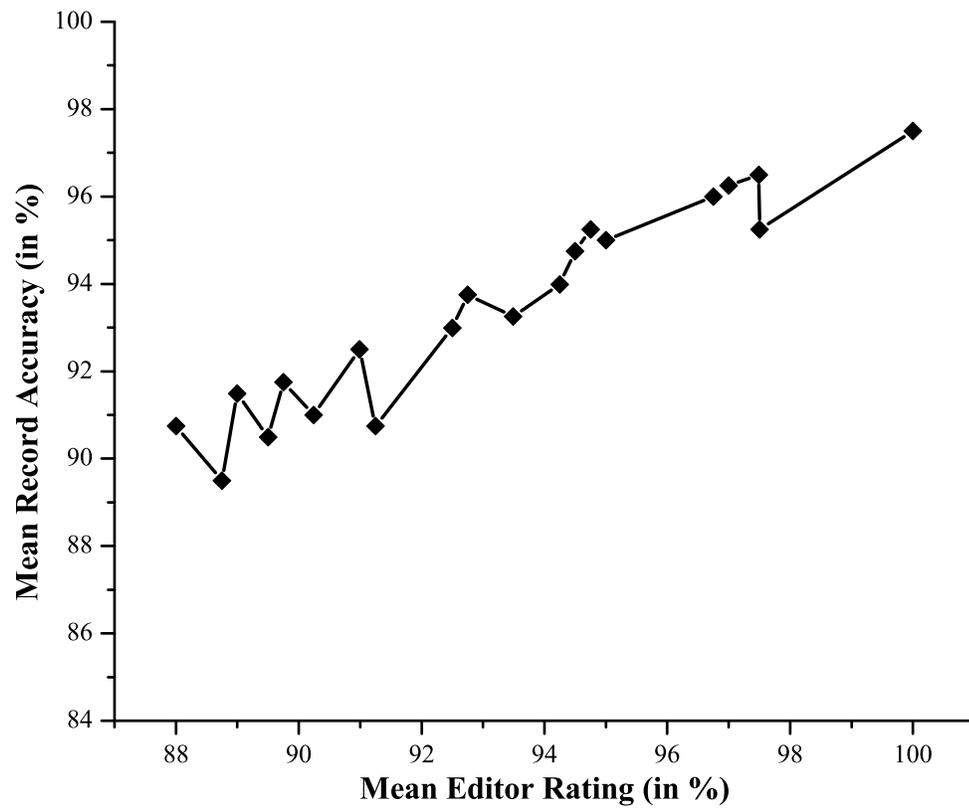


Figure 4.3: The accuracy rating of records for moderator rating of 0.90. The x-axis is the rating for editors in percentage and the y-axis is the accuracy rating for records in percentage.

## Chapter 5

### Conclusion and Future Work

This thesis proposes an algorithm to automate the moderation process of a scientific knowledge base and we implemented this algorithm in an application called GEO. We began by explaining how the manual moderation process works in GEO. We also argued why we chose trust metrics to automate this process. Then, we discussed how trust metrics can be used to compute the trustworthiness of users. We also explained how this trustworthiness value is used to accept user's contributions without waiting for manual moderation. In addition to using trust metrics for calculating the trust on contributors, we also explained the use of trust metrics to compute the accuracy of the data in GEO. Furthermore, we performed empirical analysis to understand the relationship between these two trust metrics that were used. In this analysis, we computed the minimum trust rating that users in GEO should have so that the accuracy of the records will at least be 90%.

This algorithm will go a long way in enabling the users to have a smooth, hassle free experience when contributing to GEO. Worthy users are automatically selected by this algorithm and are rewarded by automatically accepting their contributions. We plan to verify if this simple but efficient trust metric is suitable for GEO. Improvements can be made by evaluating and comparing different trust metrics as and when real user data is

## *Chapter 5. Conclusion and Future Work*

available.

Combining the data accuracy ratings that users provide, to the ratings that are provided to other contributors for their contributions will make this trust metric in GEO extremely powerful. If these two trust metrics are inter-linked, then the accuracy rating that a data gets will affect the trust rating of the contributor who provided that data. The difficulty in doing this using a LAMP architecture was discussed before. However, implementing semantic web formats over the LAMP architecture to track and query the data will provide an efficient solution for this problem. Semantic web formats adds structure to the existing knowledge base using an RDF schema. This provides a means for users to rate a source of information or any other entity (the user who contributed, in this case) associated with the data set. This will not only enable us to associate the user with the data he/she contributed, but also enables us to rate the source of the information itself. This opens up an opportunity to create a whole new trust metric to rate the data provided by users based on the source the information was obtained from.

# References

- [1] Berners-Lee, Tim and James Hendler and Ora Lassila. The Semantic Web. *Scientific American*, 284(5):34-43, May 2001.
- [2] R. Levien. *Advogato Trust Metric*. PhD thesis, UC Berkely, USA, 2003.
- [3] C. Ziegler and G.Lausen. Spreading activation models for trust propogation. In *IEEE International Conference on e-Technology, e-Commerce, and eService (EEE'04)*, 2004.
- [4] L. Page, S. Brin, R. Motwani and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.
- [5] K Sankaralingam, S. Sethumadhavan and J. Browne. Distributed pagerank for P2P systems. In *Proceedings of the Twelfth International Symposium on High Performance Distributed Computing*, Seattle, Washington, USA, June 2003.
- [6] P. Massa and P.Avesani Trust-aware collaborative filtering for recommender systems. In *Proc. of Federated Int. Conference On The Move to Meaningful Internet: CoopIS, DOA, ODBASE*, 2004.
- [7] P.Avesani, P. Massa and Roberto Tiella A Trust-enhanced Recommender System application: Moleskiing In *SAC '05: Proceedings of the 2005 ACM symposium on Applied computing*, Pages 1589-1593, 2005.
- [8] RAP - RDF API for PHP V0.9.6.  
<http://www4.wiwiss.fu-berlin.de/bizer/rdfapi/>.
- [9] Patrick Hayes. RDF Semantics. Editor, W3C Recommendations, 10 Feb 2004.  
<http://www.w3.org/TR/2004/REC-rdf-mt-20040210/>.

## References

- [10] Dumbill, Ed. XML Watch: Finding friends with XML and RDF. IBM Developer Workds, June 2002.  
<http://www-106.ibm.com/developerworks/xml/library/x-foaf.html>
- [11] Matt Blaze, Joan Feigenbaum and Jack Lacy Decentralized Trust Management. In *Proc. 17th Symposium on Security and Privacy*, pages 164-173, Los Alamitos, 1996. IEEE Computer Society Press.  
<http://www.w3.org/TR/2004/REC-rdf-mt-20040210/>.
- [12] Marko Rodriguez and Johan Bollen An Algorithm to Determine Peer-Reviewers. CoRR, abs/cs/0605112, 2006. <http://arxiv.org/abs/cs/0605112>.
- [13] Global Energy Observatory (GEO).  
<http://openmodel.newmexicoconsortium.org/GEOwiki/>.
- [14] Slashdot. <http://en.wikipedia.org/wiki/Slashdot>.
- [15] Wikipedia. <http://wikipedia.org>.
- [16] DBpedia. <http://dbpedia.org>.