SOCIALLY-AWARE TRANSPORTATION SYSTEMS

A Thesis

Submitted to the Faculty of Graduate Studies and Research

In Partial Fulfillment of the Requirements

For the Degree of

Master of Applied Science

In

Software Systems Engineering

University of Regina

By

Khaled Soliman Abdelbaki

Regina, Saskatchewan

May 2013

UNIVERSITY OF REGINA

FACULTY OF GRADUATE STUDIES AND RESEARCH

SUPERVISORY AND EXAMINING COMMITTEE

Khaled Soliman Abdelbaki, candidate for the degree of Master of Applied Science in Software Systems Engineering**,** has presented a thesis titled, ***Socially-Aware Transportation Systems,*** in an oral examination held on April 19, 2013. The following committee members have found the thesis acceptable in form and content, and that the candidate demonstrated satisfactory knowledge of the subject material.

External Examiner:	Dr. Samira Sadaoui, Department of Computer Science
Supervisor:	Dr. Mohamed El-Darieby, Software Systems Engineering
Committee Member:	Dr. Shahid Azam, Environmental Systems Engineering
Committee Member:	*Dr. Christine Chan, Software Systems Engineering
Chair of Defense:	Dr. Paul Laforge, Electronic Systems Engineering

*Not present at defense

ABSTRACT

The trustworthiness of information supplied to users of an Intelligent Transportation System (ITS) application is a crucial factor for the success of these applications, affecting these systems' credibility and reliability. The criticality of information within ITS applications requires extra steps to make users trust these applications in a way that they will make ITS applications part of their daily use. In this work, a software system is proposed that takes advantage of an online social network as a source of trusted information between different users. This work exploits social relationships to infer trust values between users who are directly or indirectly connected to each other. We have derived trust propagation equations that infer trust based on the social distance between users, global reputation, and different groups of user categories. Trusted information is used in conjunction with transportation recommendations made by other users to help users decide whether or not to follow these suggestions. The proposed system achieves a merger between transportation information and online social network data in a way that enhances users' experience of and trust in transportation information. Algorithms, architecture, and design were built for the proposed system, and a proof of concept for the system was implemented to show system behaviour. An evaluation of the system is made through simulations to show how such a system would behave and how performance would be affected by calculations of social trust between users. The main conclusion is that trust in ITS applications can be achieved through the calculation of propagated trust derived from online social networks.

ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisor, Dr. Mohamed El-Darieby, for his guidance and valuable support with my research work. His willingness to give his time has been appreciated.

Thank you to the Faculty of Graduate Studies and Research for the Graduate Scholarship I was rewarded, which supported me through my Master's degree program.

I wish to thank every single member of my family, especially my parents Fatmah El-sheikh & Dr. Soliman Abdelbaki and my uncle Dr. Ezzeddin El-Sheikh, for their moral support, encouragement, and love throughout my studies. Their examples of hard work and drive paved the road for my success as a master's student.

Finally I would like to thank my office mate, Mohamed Aboelela, for his insight, feedback, and willingness to make time to discuss issues that arose.
I would like to show my gratitude to my friends who contributed in one way or another in the course of this project.

Last but not least, I would like to thank Brandy Leippi for her help and support, especially with writing my thesis and reviewing its language and writing style.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure

Page

1. INTRODUCTION

1.1. Motivation

Increasingly, motorists rely on information technology solutions that guide them through their routes while commuting every day to different points of interest to them [1]. Expensive infrastructure equipment is required to be deployed on the sides of roads to help detect traffic conditions in real-time urban traffic [2].

The past few years have seen a vast increase in the use of online social networks by the general public, creating new opportunities for the software industry to create new applications [3]. The emergence of hand-held devices, especially smartphones, has introduced users to a new era of software applications [4]. Accordingly, social networks' mobile users are now more interested in different types of applications that fit more with the mobility nature.

More individuals are using smartphones that allow them to access different online resources like Online Social Networks (OSNs). OSNs enable their users to communicate on personal levels and share interests and information. This information-sharing capability can be very useful in the case of urban traffic recommendations. Users can share traffic information by using their smartphones or associating their vehicles' on board computers to their OSN accounts. Such association serves two purposes:

1. Sharing traffic information over social networks that increased numbers of individuals are using by using their smartphones as traffic sensing devices.

2. Incorporating a level of trust in shared information by utilizing social proximity between users of OSNs.

The main problem that this work addresses is to increase trustworthiness of traffic information delivered to Intelligent Transportation System (ITS) users. As trust increases in information exchanged over ITS applications, their credibility increases, which encourages users to accept ITS applications as part of their daily use.

The main goal of this work is to incorporate online social networks' trust between users with transportation systems in order to increase user confidence in information supplied to them by navigation and transportation applications. The incorporation of the indirect social trust between two social network users forms a means towards achieving this goal.

The target of this work is to build mobile software that makes use of social trust between online social network users to increase users' trust in traffic data given to them by transportation applications, thereby making intelligent transportation systems more trustworthy. Being able to take advantage of the increase in social network users and applications is a great benefit towards building social trust-supported applications.

1.2. Overview of Proposed System

The proposed solution depends basically on crowdsourcing traffic information in which users themselves are the main source of information. As users share traffic information about their trips, a level of trust in shared information needs to be achieved. Social trust is utilized to achieve trust in ITS applications. Social trust itself is derived from the use of OSNs as an emergent tool of communication between individuals.

To achieve confidence in shared data, the transitive nature of social networks is utilized by following different trust paths that are formed by links between two system users. These users could be directly or indirectly connected. Collectively, indirect trust values between a user who is asking for information (*truster*) and other users who are making suggestions (*trustees*) show the trustworthiness of recommended information.

Intelligent transportation applications and trust in OSNs are the two main components of the basic building blocks of the proposed system. The proposed solution merges both components and makes use of online social trust as a confidence factor in accepting suggestions given by intelligent transportation systems' users.

A detailed design of a complete solution was made to achieve the purpose stated earlier. A database was designed and built to hold both transportation situations information and social information. This database is used as the main source for social trust information that is used to infer propagated trust between indirectly connected users within the social network. Furthermore, if an indirect friendship connection is not detected to a third degree, a global trust rating (reputation) will be used that represents trust in recommending user's opinion.

Information processing, storage, and delivery are made possible through building basic services components. These components are responsible for all the business logic within the system and doing all the computations required for building trust guided recommendations based on users' social relationships within the OSN.

A modular software framework was built to connect all the different system components. The trust propagation equations required were built to infer indirect trust propagation between individual OSN users. These equations were used to handle the general case for trust propagation and its exceptions. Problems like multiple trust paths that have different lengths and lead to different propagated trust values are discussed. Assumptions were made to facilitate calculations and make the system software faster and more responsive.

A set of ITS applications that depends on crowdsourcing traffic information and utilizing online social trust to achieve confidence in shared data was designed and is described in the following chapters. The proposed applications solve some of the day-to-day intelligent transportation systems users' needs, either commercial or business, while travelling.

1.3. Scope

This work is an attempt to the merger intelligent transportation systems and online social networks in order to make use of and show the benefits of trust propagation in achieving confidence in shared data.

The following are not included in this work as they are beyond the scope of the project:

1. Detailed discussion of ITS components like applications, communication technologies (vehicle-to-vehicle, vehicle-to-infrastructure), and protocols.

2. Detailed discussion of how users express trust in each other, as it is a broad area and has its own research protocols and techniques.

Instead this work does include the following:

1. Discussion of how to use already existing trust ratings between users.

2. Demonstration of how existing trust ratings are used to study the derivation of propagated social trust between users who are not directly connected.

3. Utilization of traffic data fed to the proposed system by users through their handheld devices.

4. Demonstration of how to incorporate propagated social trust in ITS applications to achieve trustworthiness in traffic recommendation information.

The following assumptions had to be made in order to derive propagated social trust:

1. Propagated social trust is calculated up to third degree friendship relationships.

2. In case of more than one trust path, the shortest trust path is chosen to derive propagated trust.

3. The highest propagated trust is chosen when there is more than one trust path of equal length.

Detailed descriptions on how these assumptions are applied are given in section 4.4.

1.4. Contributions

The main contribution of this work is a methodology for using online social network data, especially inferred trust, for transportation systems to provide confidence in using shared transportation information by expressing trustworthiness in information sources. This contribution is achieved by:

- Deriving trust equations and algorithms in order to calculate propagated trust between a social network's users.

- The description of novel ITS applications that take advantage of both crowdsourced applications and the use of OSN trust.

- Building algorithms for how basic examples of ITS apps might be built with the proposed methodology. Namely, these applications are City Map, Freight Trust, Social Traffic Calendar, and Traffic Trust.

The contributions of this work offer advantages for both users and technology integrators, taking advantage of already existing technologies in the market. Advantages include but are not limited to:

1. Enhanced coverage of such a system due to the reliance on smartphones that users carry as traffic sensors and their Internet connectivity without the need to deploy Road Side Units (RSUs), especially in areas that are not easy to cover with infrastructure components.

2. The technologies utilized rely mainly on already existing wireless technologies (3G, 4G,...), which will not require special equipment to transfer data to Traffic Management Centers (TMCs).

3. As there is no need for largescale deployment of traffic infrastructure and communication components, cost will be decreased dramatically, allowing for building more cost effective ITS applications.

4. Within areas that have traffic components deployed already, the proposed applications can help enhance system credibility and reliability of information supplied to users.

1.5. Thesis Organization

The remainder of the thesis is organized as follows: Chapter 2 is a literature review of work that has been published with respect to trust computations and how trust propagates in social networks. It also shows different vehicular techniques for route guidance systems. Chapter 2 ends with a review of how to achieve trust in vehicular applications. Chapter 3 introduces an overall design model for the intended system as well as a detailed description of its main building blocks. In Chapter 4, a detailed design of the system is given outlining the functions of each component and how components interact with each other. Finally, Chapter 5 discusses performance metrics of the system and the results extracted from simulations made using different scenarios.

2. BACKGROUND AND LITERATURE REVIEW

This chapter introduces two key components in the proposed framework: (1) Intelligent Transportation Systems and (2) Trust in Social Networks. The first two sections discuss the techniques for both components that were studied by other researchers in the literature, showing the benefits and drawbacks for each approach. The third section covers efforts in the literature to study trustworthiness of information collected from vehicular applications and how this is utilized to deliver secure and trusted applications that users can rely on daily. The last section discusses how trust assignment works in social online social networks.

2.1. Trust in Social Networks

In order to build a successful recommendation engine for transportation data, each user's opinion in the suggested topic needs to be measured. With the use of online social networks (ONSs), suggestions can be gathered and analyzed from multiple individuals within the network. Some users might be direct friends and some others are indirect friends.

Many questions arise among OSN users with respect to the information they receive, such as:

- Do I fully trust these recommendations?
- To what extent can I trust them?

Trust information with respect to recommendations is important in the area we are trying addressing as transportation information is critical information that can lead certain users to make decisions based upon it that can noticeably impact their day-to-day activities and schedules. Thus, information needs to be verified and its sources must be trusted.

Thus, the main concern is to measure the degree of trust in recommendations that any user gets from friends or friends of friends or even deeper. To be able to do that, a trust equation has to be built that measures the trust in recommendations and its propagation across the social network. Many studies have addressed the recommendation equation and how to build it and have used different perspectives to build it and have been applied within different fields.

2.1.1. Peer Trust Evaluation Framework

In [5], the authors proposed a study on how to measure trust propagation and transition within a social network. Two kinds of trust associate each user to other users in the network [28]. Those two kinds are recommendation trust and attribute trust.

As stated in [5], a trust certificate is a "*well formatted digital document*" [5] that is used to identify users in social networks. Each trust certificate is issued by a social network user, which represents her trust values in other network participants. This digital document has a certain format that represents the *issuer*, *target user*, *trust category*, *trust value*, and a *digital signature*. "*A social network participant may issue attribute trust certificates to its direct neighbours and issue recommendation trust certificates to the trustworthy recommenders, which, it believes, would make good recommendations about other entities' attributes*" [5].

A trust chain is a logical connectivity between trust certificates along the trust path from the *truster* to the *trustee* to calculate the *trustworthiness* of a certain social network participant. Another definition used is the Trust Certificate Graph; this graph represents multiple paths and chains of trust certificates from the truster to the trustee, forming a graph.

After the construction of the trust graph another process takes place, which is the Trust Chain Fusion. In this process, all trust chains are fused into one virtual trust chain that is used later to calculate transitive trust.

2.1.2. Trust Inference Based on Bayesian Network

In [6], the authors identify the existence of intimate relationships between participants in the social network as affecting greatly the degree of trust and confidence in recommendations [7]. This value was defined as $(0 \leq r \leq 1)$ where r is the intimate degree of relationship. This value could be evaluated based on the type of interactions between participants in the social network or through the definition of the type of the relationship [8].

Another principle utilized by the authors was the importance of participants' roles and expertise in the recommendations. It is common that opinions and recommendations from experts of a certain domain are trusted and more credible than those from non-experts or beginners [9]. So, another value ρ has been defined as the impact of a participant's role in the trust value of the recommendation. ρ $(0 \leq \rho \leq 1)$ if $\rho=0$ means that the participant has no knowledge in that domain. The individual could have knowledge in any other domain, but this value has to be determined for the specific domain alone. Meanwhile, if $\rho=1$, it means that the participant is an expert in that specific domain. Social positions and titles given by social networks could help in defining such values for participants in different domains [10][11].

A Bayesian network formed the main representation of the set of concepts used in [12] as it represents both the causality in relationships between participants in recommendations, which were represented by Directed Acyclic Graphs; and the uncertainty in trust measurement that was represented by probability.

This study evaluated two types of trust calculations. Trust inference in a single trust path, in which a multiplication trust has been used to measure the propagated trust between truster and trustee. In the case of existence of multiple trust paths, another methodology was used in which the trust values from each individual trust path had to be different in each path; values r and ρ had to be calculated over each trust path.

2.1.3. Social Context

The authors in [13] are concerned with the inclusion of context information, especially in social networks, and what type of information in social networks could be considered in a social context. As mentioned in [14], social networks enable users to:

- Construct a public or semi-public profile within a bounded system.
- Articulate a list of other users with whom they share a connection.
- View and traverse their list of connections and those made by others within the system.

Social networks enable users to maintain pre-existing social relationships as well as make new connections with strangers [15][16]. Accordingly, relationship information that is acquired from any person's social network contacts can be considered a social context as this information could contain "who you are with" or "social situation"[17], "identity"[18], and "social relations"[19] or even "who you are similar to"[11].

In social recommendations, we need to make sure that recommended sources are credible. The credibility of the source actually depends on "trustworthiness" and "expertise" [20]. "*Trustworthiness refers to the possible bias/incentives that may be reflected in the source's information...Expertise refers to the perceived competence of the source providing the information*" [21].

Three hypotheses formed the basis for authors' idea about social network information and context and how trustworthy they are:

1. Recommendations from a person's social network contacts are perceived as trustworthy and relevant.

2. If the recommendation from social network contacts includes information about the recommending person it will be perceived as more relevant or more trustworthy.

3. The smaller the social distance between the sender and the receiver of the recommendation, the more important will the recommendation be perceived by the receiver.

According to experiments applied to test these hypotheses, they can be accepted as a basis to calculate trustworthiness of recommendations from contacts within the social network.

The authors concluded that users should enhance the information content of their social networks and add more details to it as well as part of their relationship information. This information is part of the user's social context that will help enhance other users' experience and help both give and receive more relevant recommendations on products and services. Adding more information on relationships, social network meta data, and contact information will help enhance decisions on the relevancy of recommendations made by senders. Also, the closer the sender of a recommendation is in degree of friendship to the receiver, the more relevant the recommendations will be.

2.1.4. TrustWalker

In [22], the main research problem was to characterize trust-based recommendations. This study was conducted with two primary goals:

- Getting recommendations by users in the same network of the user asking for a recommendation on a certain item.
- Getting ratings of items similar to the one the user is concerned with.

This is not an easy problem to solve since getting ratings from users in the same network might involve going deeper into the network to include friends of friends and even deeper, which is challenging. The challenge here is how deep (degree of friendship) to dig into the network to get recommendations.

Digging deeper into the network of friends has a trade-off in that the deeper one goes into the network, the more coverage one gets but the less trustworthy the results become from the distant users. The authors designed a model that combines both a trust-based approach and item-based approach.

In the trust-based approach, ratings from highly trusted close friends are expressed while those of less trustworthy far neighbours were not used. Ratings for similar items from trusted neighbours were used as well. *"The probability of using the rating of a similar item instead of a rating for the target item increases with increasing length of the walk"* [22]. Both results were combined in one equation to form a recommendation on a certain item and the degree of confidence in this recommendation.

The two major components of the system are:

1. The random walk along the trust network.
2. The probabilistic similar item selection.

The utilization of both components improves precision by not going too deep in the trust network and choosing raters at nearer distances but enhances coverage by considering similar items alongside the target item.

The proposed system in [22] is composed of several random walks through the target item and the similar ones; these random walks are evaluated, and the given rates are aggregated together through one equation to form the final rating and trust value for the recommendation in the target item. This proposed model has a feature called

"Explainability of Recommendations" [22]. This feature shows to users how ratings are calculated so users know exactly who affects the ratings of their target items.

Experiments performed on a real life data set showed outstanding performance in some areas. It was found to improve coverage of the recommendation for the currently existing trust-based approaches. It also outperformed collaborative filtering approaches that were either item based or user-based. It outperformed collaborative filtering in terms of precision as it only considers recommendations from highly trusted users. It is more useful than collaborative filtering in the case of cold starts.

2.1.5. MoleSkiing

Moleskiing [23] is a web-based recommendation system for supporting users with information about routes for ski mountaineering and their conditions. It not only supplies recommendations but gives users trust ratings for this infomation also. In this system, skiiers give their opinions about routes concerning snow conditions and weather conditions after they get back to their homes. When another user is trying to use the same route for skiing later, the user just checks the reviews about it before going to get an idea of the conditions.

The system in [23], in addition to recommending route conditions, offers trust ratings for these recommendations. These trust ratings are based primarily on direct trust statements users give to other users they trust. Besides this exclusive trust statement, there is an inclusive one for people a user might know directly to make use of their opinions with a certain degree of trust.

This inclusive trust score is inferred through trust propagation with the network of people the user might trust. "*Trust statements are weighted and range from total distrust to total trust. Trust statements are also subjective. They are also asymmetric in the sense that, if A trusts B as 0.9, this does not mean that B has to trust A as 0.9 as well*" [23]. The inferred type of trust here is called the local trust metric [24][25]. This local trust is a true reflection of the personal and subjective trust in the unknown user.

As has been noted, this personalized trust is very computationally expensive, as it requires the computation for the personalized trust score for each user individually. The authors propose that walking over the trust network would be more efficient as hops from one trust node to another through the trust network. These hops would not exceed a certain number, (e.g., 3). With the number of hops, some trust scores will be rejected if they are under a certain value (i.e., 0.6). "*The predicted trust score of user B is the average of all the incoming trust edge values, weighted by the trust score of the user who has issued this trust statement*" [23].

In order to implement the Moleskiing recommender, the authors utilized the semantic web format FOAF (Friend Of A Friend). In such a format, users export their social network information into files that express the social information they want to share with others. With the help of the extension in [26], trust information values can be added to FOAF files, expressing trust values in other users within the social network of the user.

According to the authors, Moleskiing doesnot require login information. Instead, users have their own blogs and update these blogs with the skiing information and then tell

Moleskiing that entries about new ski trips were created, making Moleskiing fetch this information from personal blogs and publish it.

2.1.6. FilmTrust

In [24], the author designed an algorithm for trust inference for movie recommendations. Two main questions were asked by the author that needed to be answered in order to reach an accurate value for inferred trust within two nodes in a social network. Those questions are [28]:

- How will the trust values for intermediate people affect the accuracy of the inferred value?
- How will the length of the path affect it?

It was shown in [28], as well as in [29][30], that accurate trust values can be obtained with shorter trust paths.

An algorithm called TidalTrust was created for inferring trust in social networks incorporating distance measurements between source and sink users. As accuracy of trust decreases with depth increase, it was suggested that instead of a fixed depth between source and sink, an evaluation for the shortest path should calculated and then used as a reference to the depth. "*This preserves the benefits of a shorter path length without limiting the number of inferences that can be made*" [28].

This trust evaluation is used in the Filmtrust application that personalizes content to users based on their trust network. Trust information for any user is gained from the user's social network that is defined using the FOAF Trust Module. FOAF vocabulary [31] forms the social network in the semantic web space. FOAF files describe users' personal information and social relationships [29]. The FOAF Relationship Module [32] and the FOAF Trust Module [29] are examples of anthologies that extend FOAF. Using the Trust Module, trust ratings can be assigned to users on a scale from 1 to 10.

According to the author, the TidalTrust algorithm showed increased accuracy in measuring trust values over the simple average ratings as well as recommended ratings generated using a Pearson correlation-based automated collaborative filtering algorithm [28].

2.1.7. Epinions

From previous studies, researchers have always tried to obtain trust ratings for a certain user via another user in the social network by walking through trust paths within trust networks for each user. This is mainly based upon previously known direct trust values of direct neighbours to a certain user. Based on these values, a cumulative approach is used to calculate the propagated trust so that a web of trust needs to be created and dug to get personal trust values across the network.

Unlike previous studies, however, the authors in [34] tried to utilize another area in social networks to get trust ratings. This technique uses a supervised learning approach that utilizes user factors like personal actions in the social network, and interaction factors that are represented by interactions between different pairs of users.

Epinions [35] is "*a large product review community that supports various types of interactions as well as a web of trust that can be used for training and evaluation*" [34]. Empirical results for this study showed [34]:

- Trained classifiers can achieve satisfactory accuracy.

- Interaction factors have greater impact on trust decisions than user factors.

2.2. Trust Assignment

After discussing how trust propagates along the trust path from the trustee to the truster and discussing the issue of having multi-trust-paths and how to choose the appropriate path, another important issue arises is how users will actually express their direct trust values to their immediate neighbours.

Many studies actually show interest in this particular issue. In [36], the direct trust values are calculated based on profile similarities between individual users within the social network. This technique is mainly based on the assumption that correlation exists between trust and profile similarity [37]. Similarity is mainly based on users having similar interests and sharing some common activities and types of interactions between them. This could include places they go, how similar these places are, the activities, like sports, they are interested in, and so on. In [38], similarity is estimated by measuring the frequency with which users communicate with each other using messaging services, with which most Online Social Networks are now equipped, or just using email service to construct users' network of trust.

Another way to generate directed trust values between users is to use explicit trust statements [39], in which each user can choose to express their opinion and trust of others in the social network. Later, these trust statements are converted into numbers that express how much trust is expressed, which can then be used to measure how trustworthy a user or a suggestion coming from that user or group of users is.

2.3. Transportation Systems

Having an effective route guidance model is an important aspect in Intelligent Transportation Systems (ITS); with such, the following can be achieved [40]:

1. Congested road segments can be avoided.
2. Efficiency of the road network can be achieved.
3. Drivers' travel time can be reduced.
4. All the above can lead to efficient fuel consumption and consequently lesser environmental effects.

According to [41], the US wastes more than 10 billion gallons of fuel and 63 billion dollars due to traffic congestion yearly. In order to reduce this waste in fuel, time, and money, an effective route guidance system is required to reduce trip and congestion times.

Route guidance systems supply drivers with traffic information and the best travel routes based on different criteria. Those criteria could be the shortest path length [42][43][44] or the fastest route [45]. To enhance drivers' driving experience and to achieve a more efficient and intelligent transportation approach, many technologies are used. Technologies exist both on the infrastructure level for transportation data collection and in the algorithms and methodologies that make use of collected data from such infrastructure equipment.

The following are some studies that addressed both approaches in order to achieve more intelligent and efficient transportation networks.

2.3.1. Dynamic Route Guidance Using Artificial Neural Networks

In [46], the authors proposed a new technique that mainly depends on the coupling between Time Recursive and Artificial Neuron Network (ANN) approaches. Each one of these methods has an important role in the route choice. Time Recursive method is mainly used to calculate different sections of the path in continuous time and select optimal route information, while to analyze portions of the road that have passed and record conditions of these portions using ANN supervised learning [46].

This approach depends on analyzing the knowledge base for route information and previous conditions side by side with the real-time information. Information is processed into several alternative route suggestions to the driver. The driver might choose to follow one of the suggestions or drive his or her own route. Actual route information is then fed back to the knowledge base to be part of the route history and will be used later on in further suggestions and predictions of road conditions.

Some drivers might not follow the suggestions given to them by the system due to several reasons, such as previous experience with the type of road conditions or the location. Thus, further path selections will depend on a full comparison between real-time information and experience.

2.3.2. A Real-Time Wireless Route Guidance System

The authors in [47] predicted travel time for routes from source to destination. Combining this prediction with the real-time feed coming from wireless communication devices fixed on cars, an estimate of the best route to travel was made and suggested to drivers.

In order to study the efficiency of the proposed technique, three different algorithms were discussed and compared:

1. **Shortest cost algorithm**: The most important target here for the route guidance system is to find the least cost path no matter what factors are affecting the algorithm.

2. **Travel Time Prediction and Estimation algorithm**: Prediction and estimation of travel time has a direct relation to avoiding traffic congestion. As an application of Support Vector Machine (SVM) theory, Support Vector Regression (SVR) [48][49] was used to predict travel time. According to [50], vehicle speed, traffic flow, occupancy, weather conditions, and traffic incidents have a close relationship with the computation of travel time.

   There is a correlation between the future value of travel time and the historical dataset [47]. According to [51], travel time can be divided into 3 categories: historical, current, and predicted, and there are linear relationships existing between them.

3. **Prediction-Based Real-time Fastest Path (PRFP) algorithm**: According to recent advances in real-time traffic data monitoring, it is possible to obtain reliable traffic information as it happens that is highly accurate. This, in turn, leads to the ability to calculate accurately travel time based on road congestion and current road conditions as reported by traffic equipment fixed on roadsides or on vehicles.

According to tests and simulations performed on each method, PRFP achieved a large level of trip time reduction and helped reduce congestion on different routes, and with the presence of a large training dataset, the system can help distribute traffic across all roads in the area.

This study and its experiments are based on the infrastructure-based approach, in which there is a main server(s) that receive a real-time information, does the processing, and sends route information back to vehicles through wireless modules fixed in vehicles (GPS, PDAs, Smartphones). Real-time traffic information is collected through roadside sensors and communication equipment installed to monitor traffic in each road.

2.3.3. Vehicle-to-Vehicle Real-time Routing (V2R2)

Unlike the architecture of the route guidance system in [47], which is mainly an infrastructure-based system, in [52], the authors propose an infrastructure-free approach. This approach mainly depends on communication and information transfer from one vehicle to another through V2V links, which are links between vehicles, utilizing wireless communication techniques like ad-hoc networks, Vehicular Ad-Hoc Network (VANET).

In this technique, the source vehicle sends a Request Query (RQ) over the shortest path, through relaying messages from one vehicle to another over that path until it reaches a vehicle in the destination. The travel time for the request over the shortest path is calculated. After that, a Route Reply (RR) packet is sent back from the destination node to the source node. The RR packet is aimed to flood all the available routes that the vehicle can travel and costs less travel time than the shortest path. The RR message is broadcast from any vehicle on the route to other potential vehicles.

By the time the RR arrives at the source node, all speeds of traversed segments are recorded within packets until it arrives at the source node for all potential routes. This way the source can easily define the quickest path to the destination. Another algorithm that is used called a V2R2 detour algorithm was used to help bypass void areas.

Performance evaluation took place using SUMO and MOVE simulators. Simulations showed that the V2R2 technique achieved 70% reduction in travel time as it helped find several potential routes that are faster than the shortest path.

2.3.4. Adaptive Route Guidance

In [53], the authors proposed a technique to build an Adaptive Route Guidance mechanism based on travelling users' feedback. This technique operates through a learning process that makes use of users' feedback on route suggestions based on the learning process aiming to deliver more convenient and satisfying suggestions to users.

This system is based on autonomous navigation, in which the car is independently functioning; it receives only supported navigation and traffic information as candidate travel paths are built within Traffic Information Centre (TIC), and then, this information is downloaded to cars subscribing to the service.

The Adaptive Route Guidance system mainly depends on three components:

1. A search task for a route that best fits the driver's preferences utilizing decision trees.

2. A measure of performance that gives an idea to what extent a suggested route corresponds to the driver's preferences.

3. An experience measure that combines both route and trip characteristics with the route choice.

In order to make route choices and personalize them to user preferences, the C4.5 DTL algorithm that was developed by Quinlan [54] was used for this purpose. To go through this algorithm and partition data until the final classification and decision, some attributes are required.

Seven route attributes were used for this purpose: Travel Distance, Travel Time, Directness, the Number of Turns, Travel Time Reliability, Types of Nodes, and Familiarity. Based on these attributes, decision trees are built and route links are chosen to form a final route path. This route is given alongside other alternatives to the user. Feedback is collected automatically from the user inclusively, depending on whether the user fully follows the suggestion, partially follows it, or does not follow it.

User feedback is an interpretation and analysis means to identify the user's reaction to supplied information and suggestions. Based on these values, a correction and a learning process is triggered to learn more and adapt to user preferences to introduce a more satisfying suggestion. As mentioned, the feedback process is inclusive, and it is not exclusively collected from users so making the usage experience effortless for users and more convenient.

2.4. Trustworthy Vehicular Environment

Trustworthiness in vehicular applications and their data is a very critical issue for users, as users need to have some level of trust in information given to them. Many sources of information in vehicular applications are incorporated to support many types of applications. The emergence of smartphones and hand-held devices helped shifting users towards sharing information and using mobile vehicular applications. So, in order for users to use these applications, users needs to be confident in information shared over these apps and trust their sources. The following is a review for trust incorporation in vehicular applications.

2.4.1. Vehicular Trustworthy Applications

In [55], an attempt to increase users' confidence and trust in vehicular applications was made. This was accomplished by making users increase application usage and utilize different tools from portable devices (Smartphones, PDAs, etc.) to on board sensors and the vehicles' computing capabilities. As stated by the authors [55], context sharing between vehicles is an important aspect through which users share real-time context information in order to suggest and warn others about certain situations.

According to [55], users will trust a vehicular application if it is available, reliable, has built in protection against modification and impersonation, and protects users' privacy. In order for the authors to achieve their goal, they incorporated the usage of certificates that increase trust through increasing security and protecting their applications from intrusions. Also, this achieved privacy for users' context information while still being able to share this information to support decision-making through these applications.

The same concept was utilized in [56]. Having a security middle-ware to secure the context database is a core concept in the architecture suggested. This security middle-ware achieves:

- Privacy and security for the context information and public key management.
- Certificate management to ensure confidence in data and information given to users of the system.

This, in turn, makes users trust the system and improves the credibility of its data.

2.4.2. Opinion Piggybacking

This technique is used in VANETs, in which opinions about forwarded messages are appended to these messages by the forwarding entity [57]. A decision whether or not to trust the forwarded message is made based on other attached opinions to the message. Three fields form the attached opinion to any forwarded message: one field represents trust in the message, another field is for trust in the message generator, and the third field is for its own ID. Both trust values in a message and its generator are based on all opinions attached to the message combined with trust values for opinion providers that are locally stored. This calculation mechanism takes the form of a cluster, where the cluster head is the entity that decides whether to forward a message or not to other nodes.

Trust threshold in [57] dynamically takes into account the spatial distance between message source and deciding entities and their familiarity with the area covered. If an entity's opinion on a message leads to a correct decision, other entities' trust in the opinion provider entity increases, and if not, then trust in the opinion provider entity decreases.

3: OVERALL SYSTEM

In this chapter, the framework of the proposed socially aware ITS system is introduced, as along with the system model and its main building blocks. Both the social and the transportation components are discussed, detailing the interaction means between the two main components of the framework.

3.1. System Requirements

In order for the socially aware ITS application to function as proposed, the following is a list of requirements that needs to be achieved:

- Users access traffic condition information in real-time, which provides up-to-the-moment information about road and driving conditions in order to enable them to plan their trips wisely as a convenient service.

- Supplying real-time information to users is supported by trustworthiness in data that is inferred from their social relationships with other users in the system.

- The dynamicity of social relationships directly affects trustworthiness of information supported by social trust. Trustworthiness is affected not only by closeness of users and directed trust ratings between them, but also by the number of users supplying information. The more suggestions given by close friends, the more accurate trust ratings are as data.

- If closeness of relationship up to third degree is not available, a global trust rating that each user gains from each successful recommendation is used to guide user recommendations.

- Users must be able to give feedback about traffic conditions in real-time whether using their hand held devices or using built in computing units in their vehicles.

- Simplicity of use of the application in order not to impair drivers' driving capability is imperative.

3.2. Solution

This section shows the solution architecture for the intended system. This solution architecture goes beyond the software and business layers to offer a complete architecture for the intended application for future implementation. Figure 3.1 shows a detailed graph for all required components and how they are layered together to form the overall system.

As shown in Figure 3.1, the system incorporates a multi-tier architecture that achieves modularity that helps make system management, maintenance, and customization easier. In this work the approach to the implementation depends on having the following:

(1) *Central server* that is performing all the business logic on the traffic and social data; this layer of servers could be implemented either using a cluster of servers that achieves redundancy and load balancing, to help improve system performance and availability, or by using the new, emerging cloud solutions that make use of all the computing resources in order to reduce overall implementation cost and achieve better utilization of resources.

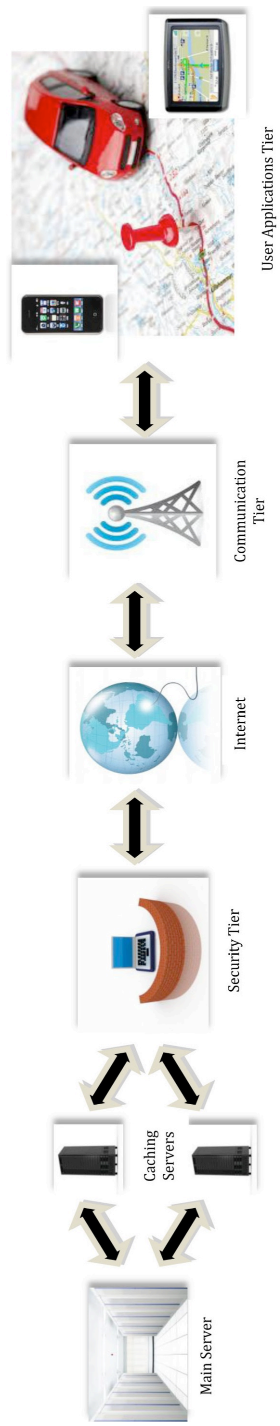

Figure 3.1: Complete Solution Architecture

(2) *Caching servers* are important in this application as they enhance delivery rate of information, which also helps improve redundancy of components and system availability.

(3) *Security tier* is required as the nature of the proposed solution is critical in many ways. Part of the information exchanged is private as it carries user information and social relationship information, as well as traffic and location services information. For this reason, the *security tier* is required to enhance system security from external intruders who are trying to get unauthorized access to this private information. This *security tier* should include some basic components to achieve the required level of security. A firewall, either hardware or software, is among these components. Also, an intrusion detection mechanism is highly necessary to prevent unauthorized individuals from accessing the information repository. This helps prevent corruption of the data that could lead to inappropriate results and prevents unauthorized users from gaining access to private users' data.

(4) *Internet layer* is required since the system relies on the Internet as the main medium for information exchange. Alternatively, a private communication network could be used. This would enhance security and reliability but would greatly increase implementation cost as it would require deploying huge amounts of infrastructure components. This can also be achieved by connecting vehicles' on board computers to the Internet, and an increased number of users now have access to the Internet through their smartphones.

(5) *Communication layer* shows what technologies could actually be used. Cellular data and GPS networks can be used to achieve the required communication with the application and web servers.

(6) *Portable devices* that end-users utilize to make use of the offered applications and services. These could be mobile web browsers (Chrome, Safari, etc.), smartphone applications for different platforms (Android, iOS, etc.), on board vehicle computers that are connected to the Internet, and GPS devices that can be connected to the Internet to enhance users' experience and make use of the benefits of both the Internet and GPS satellites.

Figure 3.2 is a UML deployment diagram that models the physical deployment for the system in Figure 3.1. It shows all nodes, artifacts, and components required to build the system.

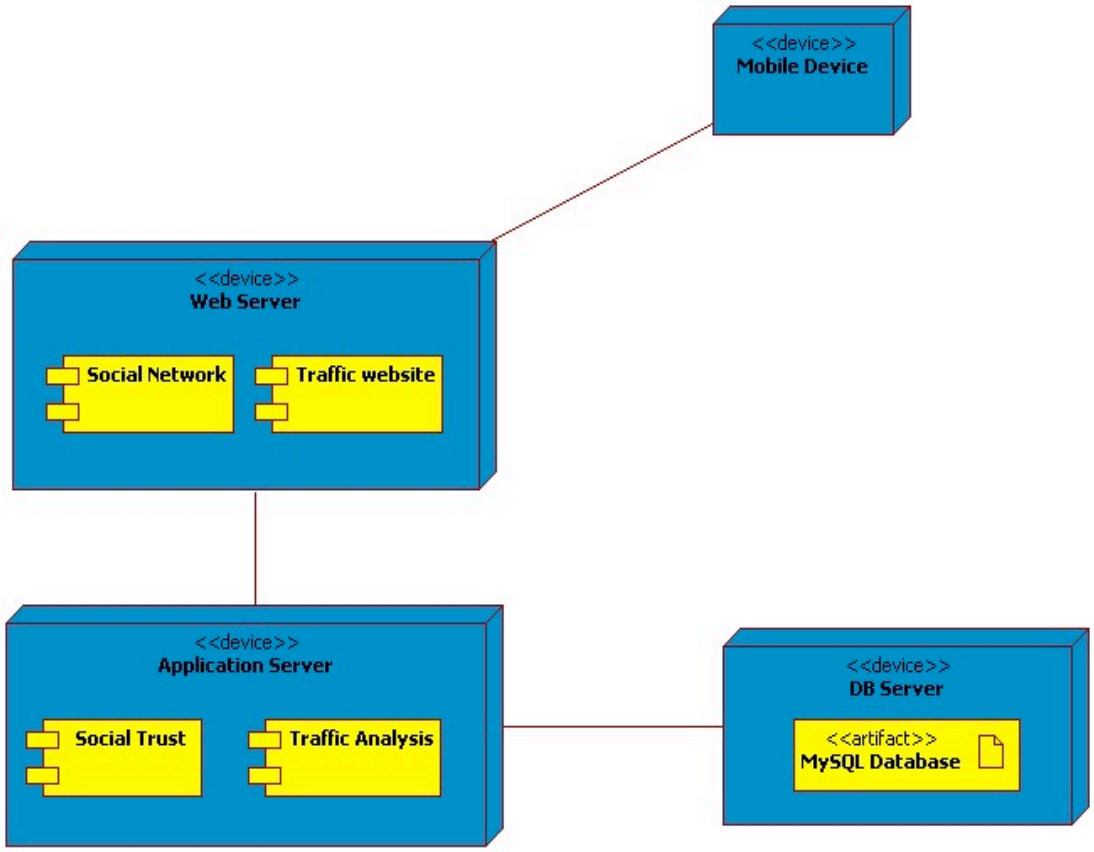

Figure 3.2: UML Deployment Diagram

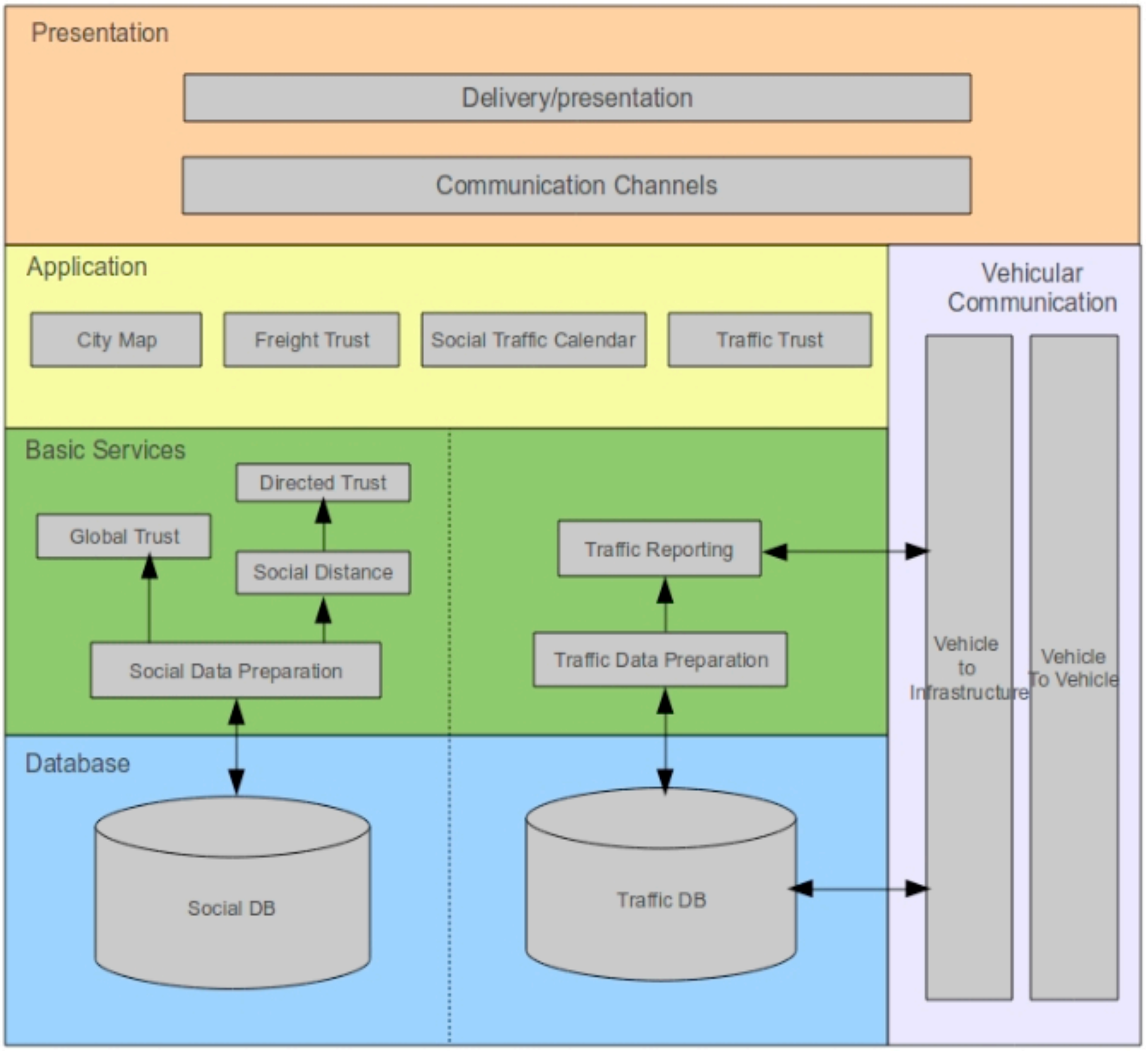

Figure 3.3: Main Building Blocks of the System Architecture

3.3. System Architecture

Figure 3.3 shows a high-level block diagram of the main building blocks of the system architecture and its abstraction layers. It mainly consists of five layers, namely the data, communication, basic services, application, and presentation layers. Each of these layers has a very specific function in the overall framework to achieve modularity in the system. Modularity eases communication between layers and makes future modifications and enhancements within each layer a simple task.

Each layer consists of several building blocks; these blocks show the function of each layer and how it interfaces with its lower and upper layers in order to exchange information between them.

The following is a detailed description of each layer and its main building blocks. The explanation starts from the bottom of the layered structure and progresses up through the architecture.

3.3.1. Data Layer

This layer forms the main data layer in the architecture, where all data elements are stored. This data is used both for live processing and for historical and analytical purposes. It mainly consists of two components:

1. The social network tables, which store the basic information required about all social network users and their relationship information. This includes direct friends for each and every user who uses the network and trust information, including direct trust information between each pair of friends.

2. The traffic (vehicular) tables that include all the information concerning traffic situations. This includes conditions that might lead to traffic delays, such as accidents on different road segments, traffic congestion, and construction work. These traffic conditions are reported by users to the system and stored for live queries and/or historical processing of data. As social network users report traffic conditions, vehicle identifiers are mapped to users' identifiers. Mapping users' identifiers to their vehicles' IDs adds a layer of separation to achieve privacy.

3.3.2. Communication Layer

This layer is concerned with handling vehicular communication. It enables users to report a certain situation that is taking place on a specific road segment and consequently retrieve road condition recommendations from other users in the network. Traffic data retrieval is supported with social trust rates in order to make a decision on how trustworthy the data is.

The communication layer is crucial as it handles real-time traffic data. Data handling happens in two forms:

1. Data collection from users in real time and storing data in the database.

2. Delivery of processed information to users enquiring about recommendations supported by social trustworthiness.

In the case of data collection, the communication layer delivers traffic data to the basic services layer that prepares the data and then delivers it to the database. While in the information delivery process, data processing happens on the raw data in the database by the basic services layer that gets delivered to the communication layer after processing. Processed data gets delivered to the application layer to be presented to users on their mobile devices.

3.3.3. Basic Services Layer

This layer is concerned with the basic services offered to users supported by social trust information, which creates two categories of basic services:

1. The first basic service is *traffic reporting;* it handles traffic requests coming from users. These requests are analyzed to help retrieve data from the database about traffic conditions related to the request. Recommendation information is then processed to be sent to the application layer for delivery to requesting travelers. Individual travelers of the system report their own speeds while commuting. These reports are prepared to be stored in the database for use in other travelers' requests.

2. The second category of basic services in the system is the *social trust*. As users start their trips, they need to get information about road conditions for the different routes they are using. This information is supplied to them with trust information about the recommendation sources through their social connection with other users within the social network. *Social trust* accomplishes this by defining social distance between individual users, the one who recommends (*trustee*) and the one who uses the recommendations (*truster*). *Social trust* calculates the *propagated trust* between individual users up to the third degree of

relationship between the truster and all the trustees and calculates propagated trust, as will be shown in Chapter 5.

Social trust is not only dependent on propagated trust, but also on global trust of individual *trustees*. This feature is used when a social relationship between the *truster* and *trustees* up to third degree does not exist. In this case, a global trust value, which represents trust of any user in the system in this recommender, is used. This value is the same for all users who need to use recommendations from that specific recommender.

Propagated *social trust* rating is the main value that the basic services always tries to derive if possible. When that is not possible, *social trust* starts incorporating *global trust* ratings.

3.3.4. Application Layer

The following is a list of possible ITS applications that can be built upon the use of social trust to achieve confidence in crowd-sourced ITS applications.

1. **CityMap:**

   The goal in this application is to build a city map that shows speeds on different road segments. Information in this map is gathered from users of the system supported by their global trust rating (reputation).

2. **Freight Trust:**

   In this application, traffic information is shared between drivers of a freight company and among two or more different freight companies. Trust transitivity within the same company drivers and among different companies is incorporated.

3. **Social Traffic Calendar:**

   This application uses calendar events to check traffic conditions in order to estimate when the suitable time to leave is to be in time for the coming events. In this application, an Estimated Time of Arrival (ETA) is derived based on traffic information supplied by other system users and supported by social trust.

4. **Traffic Trust:**

   This is a general application that can be used on a daily basis. In this application, users consult the system for traffic recommendations before travelling to a specific destination based on information coming from social connections (friends, co-workers, family).

3.3.5. Presentation Layer

This layer is concerned with information supply to application users. The presentation layer takes care of how information is introduced to users in real time and achieves security requirements in supplying information to users.

The two main blocks of the presentation layer are:

1. *Communication channels*, which are concerned with establishing a means of communication between individual users through their portable devices or onboard platforms installed in their vehicles. Also, communication channels addresses authentication and authorization mechanisms to ensure authenticity of users and to secure their respective privacy.

2. *Delivery/Presentation* is concerned with how information is introduced to users via their devices in a convenient way for them. Visual information representation and audio commands are convenient means to present information to users in a natural way without impairing their driving ability.

SUMMARY

In this chapter, an overall view was introduced for the proposed solution for the Socially-aware ITS system. Different layers within the system architecture were discussed. An overall solution architecture was introduced.

4. DETAILED SYSTEM DESCRIPTION

This chapter offers a detailed overview of the system design. It shows the algorithms used throughout the system, database design tables, equations used to calculate propagated trust throughout the system, and details for assumptions used to calculate trust over different trust paths.

4.1. Software Architecture

This section shows in detail the software architecture of the proposed system. Software is the main part of the proposed system as it contains all methods and classes that handle traffic information as well as social information. It is also concerned with the social distance calculation, which is used to calculate propagated trust values after the analysis of all the aspects and parameters for social trust recommendations and calculation.

Figure 4.1 shows the software architecture of the proposed system. The system architecture consists of 3 main layers, each of which contains some basic components that form the building blocks for each layer. The data layer is concerned with data manipulation and information retrieval either from or to the database. The services layer is concerned with all aspects of propagated trust calculation and social relationship analysis and the analysis of traffic conditions. Finally, the user interface layer takes care of user interactions and how messages are delivered to users and how feedback is collected from them.

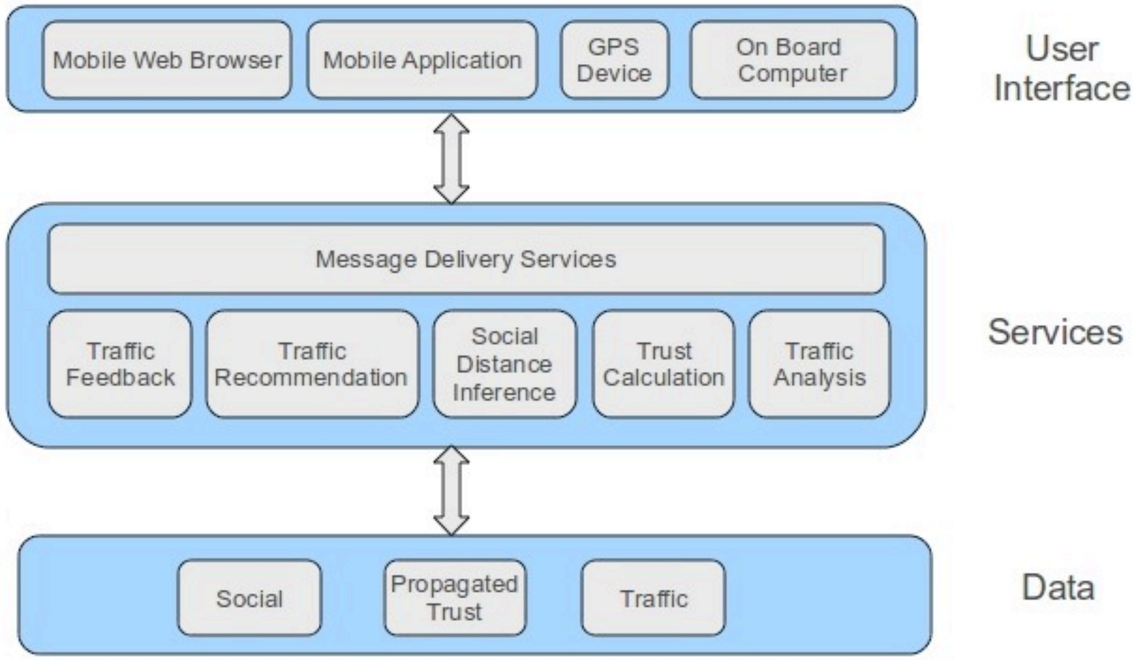

Figure 4.1: System Software Architecture

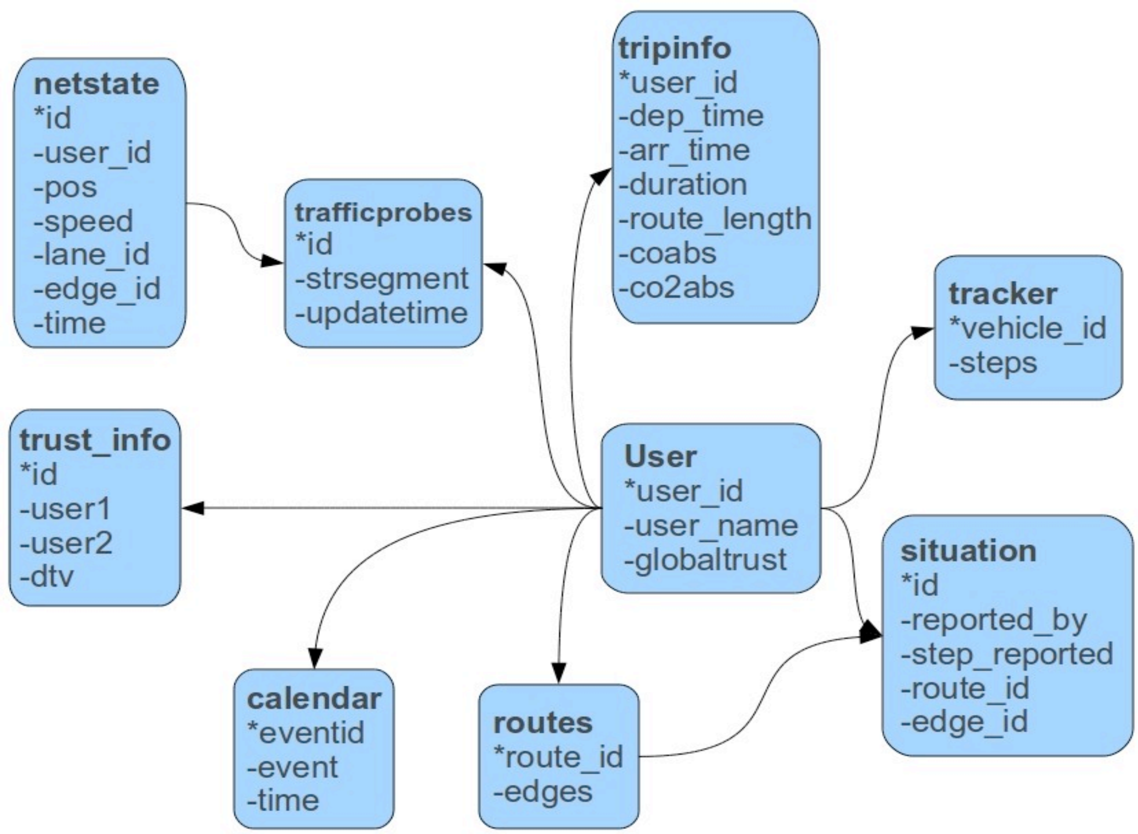

Figure 4.2: ERD of the Traffic Situation and OSN database

4.2. Data layer

The proposed system relies mainly on two main data components, traffic and the OSN. Each one has its own unique functionality through the system.

1. Traffic data includes all the information required for reporting a situation on the road. Situations could be accidents, construction, traffic congestion, and so on. Also, the traffic database contains recommendation information that will be later given to users who are interested in getting route information. These recommendations will be guided by trust ratings of the users making the recommendations.

2. The OSN data offers friendship information showing which users are directly connected and direct trust values between individual pairs of users. OSN will be used to infer trust relationships beyond the first degree of friendship, and using these direct trust ratings, will calculate the propagated trust rating in *trustees* making recommendations. Figure 4.2 shows a combined ERD for both data components.

Two different technologies can be used to implement the data component:

- Relational Database Management System (MySQL)

- Map/Reduce (Hadoop)

Each technology has its own benefits; further discussion is given in section 5.2.2.

The following is a detailed description of each table in the ERD in Figure 4.2. The description starts with tables of the social component of the database, followed by the traffic component.

4.2.1. Social Data

- `User`: contains all social network user information (`user_id, user_name, globaltrust`). The first field is a unique identifier for each user in the system. `User_name` holds the names of individual users. `Globaltrust` field holds reputation information about each user; users gain or lose reputation based on useful or useless recommendations respective to other users within the network.

- `Trust_info`: this table contains directed trust values between each pair of users. Directed, here, means generally that trust is not the same in both directions of a friendship. Both users can have different trust values in each other, or both users can express the same trust in each other. The field Directed Trust Value dtv expresses this value.

- `Calendar`: contains appointments for a certain user that are used for the social calendar applications. Each calendar event is defined by its location and start and end time.

4.2.2. Traffic data

- `Routes`: all edges that form a route are held in this table for all travelled routes.

- `Netstate`: this holds all the information about travelling vehicles (`pos, speed, lane_id, edge_id, time`) during the whole trip each vehicle travels. This table contains reports that vehicles share during any given trip.

- `Trafficprobes`: this table is used to store processed information about different road segments after processing data in the `netstate` table.

- `Tripinfo:` contains all trip information (`dep_time,arr_time, duration, route_length, coabs,co2abs`). This forms statistics about each trip made, including travel time, length, and environmental impact (emitted CO and CO_2) information.

- `Traffic_Situation:` this table holds all situations reported by motorists (accidents, congestion, etc.), this information later on will be reported to other users asking for route alternatives to travel to avoid such situations.

- `Tracker`: this is a utility table used to automate simulations and communication between SUMO simulator and interaction classes. It is useful to track different traffic situations and information on if they are already being cleared or still exist on the road.

4.3. Basic Services

Communication services between vehicles on the road and the central database is one major function that is offered in the basic services layer. This type of communication happens in two directions.

- One direction is from vehicles (motorists) to the database to report different traffic and road conditions represented by a system service called `reportTrafficEvent()`.

- The other direction is when a user is requesting a recommendation from the system through a service called `getAlternatives()`.

Another main component of the basic services is a social distance calculator through a system service called `getSocialDistnce()`, and, based on this system, propagated trust is calculated through service `getTrust()`.

The third group of basic services is concerned with analyzing user suggestions and extracting social information associated with each user making/receiving a suggestion, this group of services includes many service calls that offer such functionality. Figures 4.3 and 4.4 show sequence diagrams for operations involved when a certain situation takes place, a car accident in this case. Figure 4.3 shows a sequence diagram in case of a car accident reported by a user travelling the same road segment of the accident. It explains actions involved in the situation report case.

In case another user is travelling the same route and asking for recommendations, Figure 4.3 shows all the actions involved and required by the software in order to supply the enquiring user with a socially-supported decision.

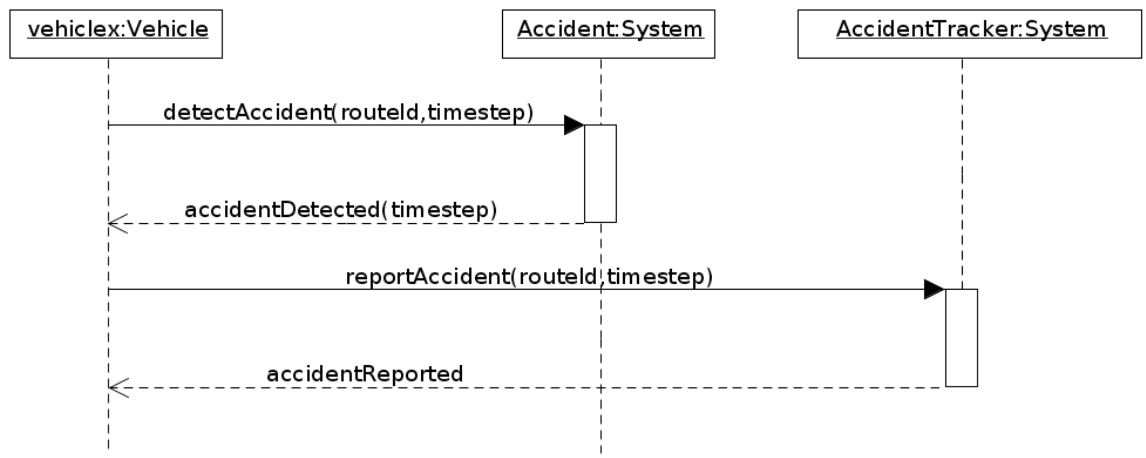

Figure 4.3: Accident Report Situation Diagram

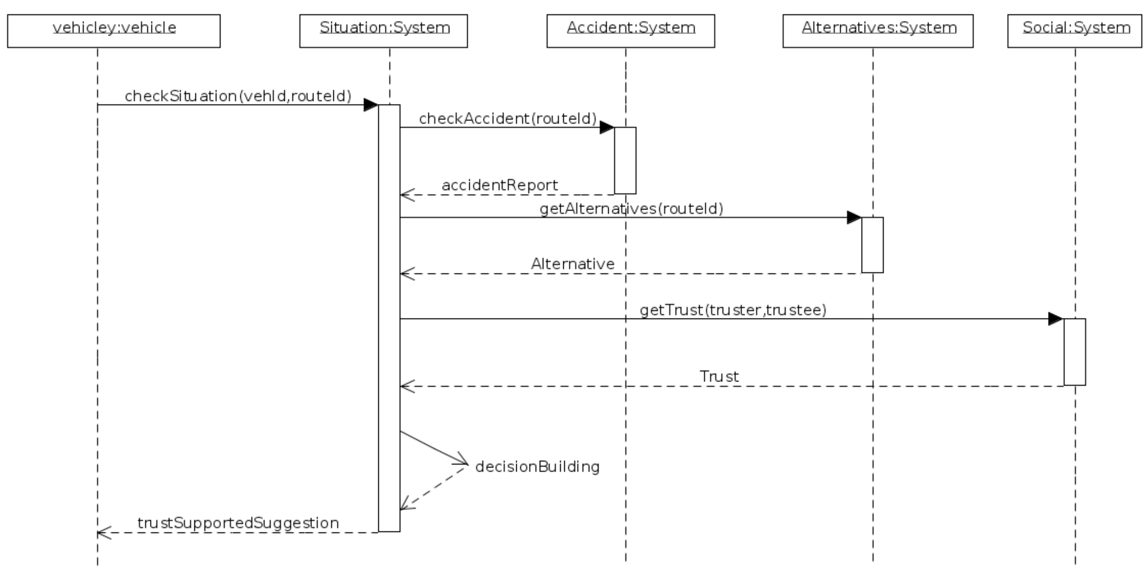

Figure 4.4: Socially Supported Route Suggestion Situation Diagram

Traffic Trust application discussed in section 3.3.4 is exactly the case discussed in UML situation diagrams, Figures 4.3 and 4.4. In this case, when a user is about to travel to a certain destination, the user usually checks the application about road conditions. This process involves checking traffic speed and conditions on all segments of the road. When a situation that is reported by another user is detected, the propagated social trust is calculated to show to what extent information coming from that user should be trusted. Supported by social trust, a recommendation is made to the enquiring user about alternative routes.

Figure 4.5 shows a pseudo code for an algorithm that checks the system service for any events that could occur on different routes between origin and destination points. As shown in the pseudo code, the algorithm then checks for recommendations for alternatives, and the trust ratings of recommenders then tells the user by how much other users' recommendations should be trust.

Basic services are part of system interaction classes. These classes fall into three groups, each serving a different purpose:

1. One group of classes is responsible for social analysis that includes social relationship information and social distance between OSN users. Social classes analyzes social relationships to measure the social distances between pairs of users, whether it is first, second, or third degree, and consequently it calculates propagated trust values.

```
Inputs:Origin point, Destination Point
Processing:
        CheckTravelledRoute()
        {
                //This method is used to query the database for different
                traffic situation for the travelled route.
        }
        CheckAlternatives()
        {
                //check the database for alternative routes between the origin
                and destination points.
        }
        CheckSocialTrust()
        {
                //check social trust of recommenders.
        }
Outputs: Social trust guided recommendation.
```

Figure 4.5: Alternative Route Algorithm Pseudo Code

2. The second group of classes is responsible for accepting inputs from OSN users (motorists) about traffic conditions and reports and stores this information in the database. Traffic information is made available for other users who enquire about traffic conditions on a real-time basis.

3. The third group of classes is concerned with the interaction and links between traffic suggestions and social network trust. It is also concerned with how to deliver the data to users enquiring about traffic conditions. Thus, this group is responsible for the merger between the OSN and traffic system that achieves the main target of this work.

4.4. Trust Propagation

The calculation of *propagated trust* is one of the most important factors in the experiments, and it is a challenging task. Usually it is not a problem to get a trust value for first degree friends, as this is stored in the social network data tables. However, for second and third degree friends, more work is involved to infer the social distance between users. The difference in inferring social trust for second and third degree friends is that a friendship relationship could have multiple trust paths.

Figure 4.6 shows an example of multiple trust paths between two individual users. The trust path between two nodes, **a** and **b**, has two alternatives, one of which passes through nodes **x** and **y** and the other that passes through nodes **i** and **j**. When calculating the propagated trust values between the two main nodes, we would get two different values following each path.

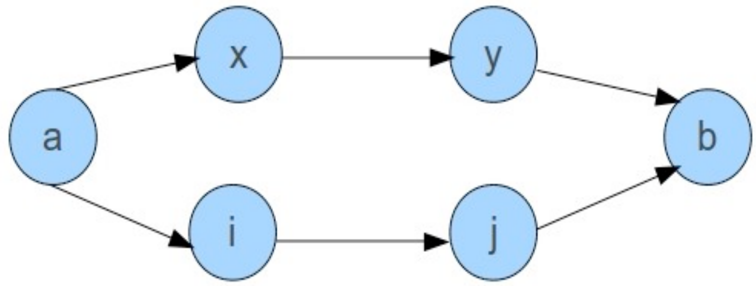

Figure 4.6: Multiple Trust Paths

A decision has to be made to choose between the two values obtained. In this work, we decided to go with the highest trust path values between the two main users. So, if the trust along the **xy** path is higher than that of the **ij** path, then the algorithm would choose the **xy** path.

The calculation of the propagated trust value itself along the trust path is done by calculating the average trust value of directed trust values between each pair of users over the trust path. For example, if we want the trust value between **a** and **y** in Figure 4.5, the average of the directed trust between (**a**, **x**) and (**x**, **y**) pairs is calculated, and that average forms the propagated trust between the (**a**, **y**) pair.

To simulate this trust propagation, node (**a**), which needs to know the trust rating in a certain node (**b**), just asks all its adjacent nodes (neighbours) how much they trust this specific node, and in turn, those adjacent nodes ask their adjacent nodes until they get feedback about the trust rating along with how deep the relationship is. Trust rating and depth of relationship are all given to the enquiring node in order for it to decide how much it can express trust in the trustee.

The example of trust propagation between nodes **a** and **y** works only with inferring propagated trust in the case of one single node making a suggestion. Concerns arise when more than one node is involved in the suggestion process as overall propagated trust calculation in all trustees becomes a more complicated task. To address these concerns, the simple average between each pair of nodes, the truster and each single trustee, still applies.

To get the overall propagated trust, a weighted trust equation is designed as shown in Equation 4.1.

$$T_{overall} = \frac{\sum W_p T_p}{\sum T_p} \qquad (4.1)$$

Equation 4.1 calculates propagated trust as a weighted value. Each degree of friendship participates with a weight in the overall trust according to its degree of friendship. Accordingly,

- First-degree friends participate fully with their trust values.
- Second-degree friends participate with half their propagated trust ratings.
- Third-degree friends participate with one third their propagated trust rating.

This was the general case for calculating the overall trust, when more than one trustee is involved in the decision making process.

Equation 4.1 has two exceptions: Equation 4.2 shows one exception in which the numerator and the denominator are equal, which will result in a trust value that is equal to 1. This value is not realistic as this case happens when there is one and only one trustee and it is a first-degree relationship. So, if trust in this user is less than 1, the overall trust cannot be higher than that value, so we chose to make it equal to that trust value in this specific trustee.

$$if \sum W_p T_p = \sum T_p, T_{overall} = \sum T_p \qquad (4.2)$$

The other exception with Equation 4.1 is when the denominator equals zero, which will lead to a division by zero error. In such a case, we can simply substitute zero for the trust value as this value means nobody is suggesting anything to the truster so there is no trust value to use. This is shown in Equation 4.3. This situation occurs only in the case of a user who just joined the social network and does not have any friends yet or is still in the process of connecting to their friends (Cold Start).

$$if \sum T_p = 0, T_{overall} = 0 \tag{4.3}$$

In all three previous equations, $T_{overall}$ represents the overall propagated trust of the *truster* for all *trustees* participating in route alternative suggestion. $\Sigma W_P\ T_P$ represents the summation of all weighted ratings for trust values for all levels of friendships, up to third degree. ΣT_P represents the summation of all trust values of all three levels of friendship.

Another concern that arises here is when we have more than one trust path between the *truster* and each *trustee* that are also different in length; this situation is represented by Figure 4.7.

Figure 4.7 shows a case in which two nodes that are not directly connected have multiple paths, each of which has different social distance. We have two paths between **a** and **b** (**a→x→b**, **a→i→j→b**). In the proposed algorithm, we decided to use the shortest social distance path between the truster and the trustee, which, in this case, is (**a→x→b**). This choice offers a faster calculation for the propagated trust value by reducing the need to go deeper within the social graph.

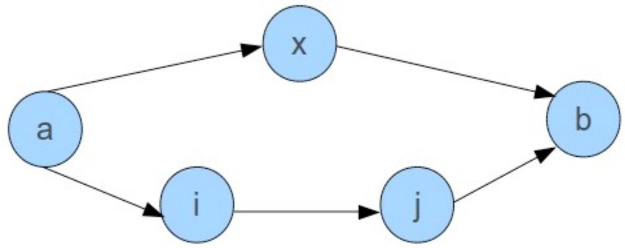

Figure 4.7: Shortest Path to Destination

Additionally, propagated trust relies mainly on trust values between each pair of adjacent nodes, which constitute the overall trust rating, so the shorter the social distance, the better the propagated trust.

To summarize, two main factors greatly affecting propagated trust can be identified:

1. The social network topology, or, in other words, who is connected to whom within the social network. Topology changes will affect propagated trust.

2. Directed trust values in which changes in these values will lead to a change in propagated trust ratings and could possibly affect it in a way that increases or decreases the overall trust.

4.5. Trust Algorithm

In the previous section, trust calculation equations were shown for different situations and exceptions. In this section, the algorithm for these equations is described. Figure 4.8 shows the pseudo code for the trust calculation algorithm.

The pseudo code in Figure 4.8 does two main tasks:

1. Sorts and combines *trustees* into separate arrays, each of which corresponds to a degree of friendship (first, second, third). This process involves calculating social distance between the *truster* and each *trustee* using *checkDegree()* method.

   After combining *trustees* by degrees in separate arrays, a trust rating is calculated for each degree by taking the simple average of trust values in each array.

Inputs: Array of all users participated in a recommendation *trustees[]*.
Processing:

```
propagatedTrust(trustees[])
{
        For trustee in trustees: //To extract degree of firnedship
                Degree = checkDegree(trustee);
                If degree == 1st:
                        Add user to the first degree array;
                Else if degree == 2nd:
                        Add user to the second degree array;
                Else if degree == 3rd:
                        Add user to the third degree array;
                .
                .
                .
                Else if degree = nth:
                        Add user to the nth degree array;
        calculateTrust(1st[], 2nd[],..., nth[])
        {
                For each array:
                        Weighted_trust = degree_weight*array_average;
                        Total += array_average;
                Trust = weighted/Total;
                Return Trust;
        }
}
```

Outputs: Overall propagated *trust.*

Figure 4.8: n-Degree Trust Calculation Algorithm Pseudo Code

2. Calculates overall propagated trust using *Directed Trust Values* and *calculateTrust()* method utilizing Equations 4.1, 4.2, and 4.3.

SUMMARY

In this chapter, a detailed description of the proposed system that was introduced in Chapter 3 was discussed in detail. Basic services performed by the system, data management, and software architecture were explained. UML diagrams for possible situations and how the system handles data were shown. Trust propagation as a very important problem to address in this work, and it was explained in detail, along with its exceptions and possible solutions. Also, possible algorithms for trust propagation were proposed, explained, and shown in pseudo code.

5. SIMULATIONS & PERFORMANCE ANALYSIS

A simulation model was built and utilized in order to simulate social interaction between motorists on a real-time basis and how that affects suggestions supplied to them by the software framework. The simulation model also gives an idea of how efficient the application of online social networks might be as one of the tools available for ITS applications.

In this chapter, a detailed explanation of the simulation model incorporated in this work is discussed and analyzed.

5.1. Simulation Objectives

To measure performance for the proposed system, some goals and objectives are targeted to have a real and better understanding of how the system is actually behaving. Two main values are to be assessed through all simulation runs:

(1) *Propagated trust* is defined as the calculated *trust* through the propagation of direct trust values of the intermediate neighbours of both the *truster* and the *trustee*. How the depth of a relationship affects trust propagation through the usage of Equations 4.1, 4.2, and 4.3 is measured through experiments.

(2) *Responsiveness* of the system represents the time consumed to process a user's request for an alternative route guided with a *social trust* rating. This value is measured in cases of *global* and *propagated trust*. In the case of propagated trust, it consists of two values, walk time and trust calculation time. The walk time is the time taken to calculate the social distance between both *truster* and *trustee*,

while the trust calculation time is the time taken to use one of the trust equations based on the distance to calculate the *propagated trust* value.

5.2. Simulation Model

In this section, the simulation model is discussed with all its major components and the interactions between them. The simulation model has, in fact, the same architecture as the software framework presented in Chapter 4, including all the layers and the main building blocks within each layer.

The simulation model consists of four main building components that represent all five layers in the software framework. These four components are:

- SUMO Traffic Simulator [57].
- Online Social Network.
- Traffic Situation Information.
- Real-Time processing/interaction classes.

The following is a detailed description of each one of these components.

5.2.1. SUMO Traffic Simulator

As shown in [57], SUMO is an open source traffic simulator that gives us the ability to simulate different real-life traffic patterns and road network conditions. In this work, two components within SUMO were utilized. The first component is the traffic simulation function itself that allows us to implement any scenario and traffic situation as needed. This component helps us apply different road conditions, as well as some functions that allow incorporating actual urban traffic conditions like school zones, speed limits, bus routes, and other similar features. This component makes simulations more realistic by depicting real-life conditions.

The second component to be used in the simulation model is TraCI (Traffic Control Interface). This tool helps with real-time interactions within simulations through allowing simulated vehicles to send traffic and driving conditions to the communicating tool for real-time processing of this data or accepting real-time commands from external tools to change driving conditions. So, TraCI makes simulations more interactive and dynamically responsive to changing traffic conditions. Figure 5.1 shows the sample road network that has been used throughout the simulations.

5.2.2. Online Social Network

The Online Social Network (OSN) is a main component in simulations and a core part in the software framework, so simulation for an OSN was required. In order to simulate an OSN and incorporate it into the simulations, the following components were required:

- Users in the OSN are the main part and the core component of any social network as all social interactions take place through those users.
- Relationships between OSN users are the second main component that forms OSN's social topology and who is a friend of whom and forms the social graph.
- Degrees of trust each user in the OSN expresses in their direct friends.

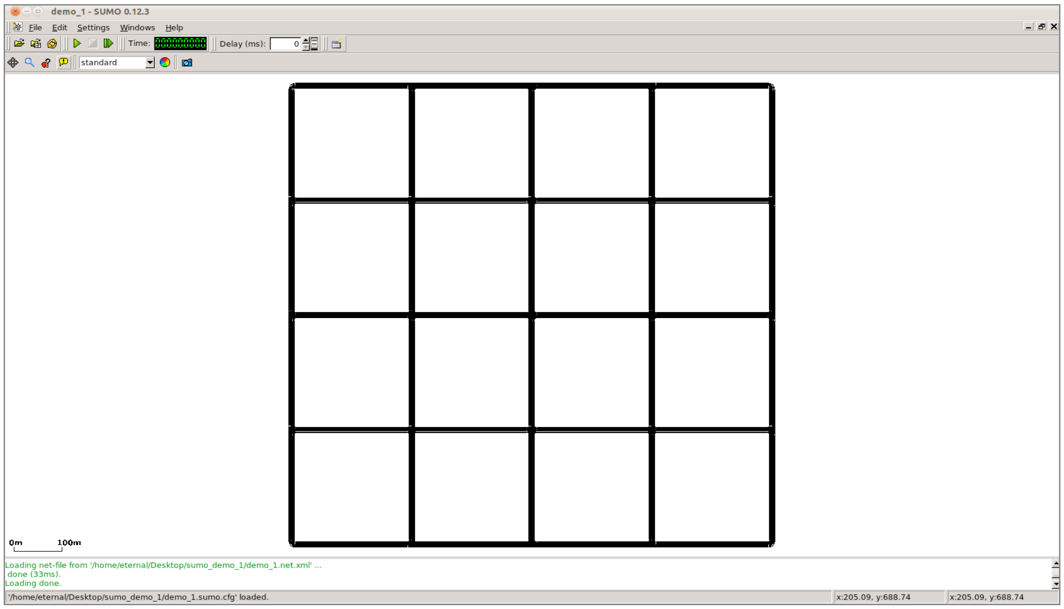

Figure 5.1: Sample Road Network

The last part is very important in simulations as it forms the main outcome from user interactions that will be directly utilized in supporting suggestions given to other users within the network.

Degree of trust between pairs of users is not a value that is known explicitly; instead, it is an inferred value. The inference of this value depends on many factors, like number of interactions between pairs of users and the nature of the interactions themselves. The process of trust inference is not in the scope of this work, and for the purpose of this study, predetermined values for mutual trust were used directly to simulate trust ratings between pairs of users.

The simulation of the OSN and its trust relationships is going to be done through several database tables. One table will contain all users within the OSN with their user IDs. Another table within the database will store all relationship information. Relationship information includes who is friends with whom and directed trust values between pairs of friends. In this way, we can dig deeper into the OSN to get relationship information beyond the first-degree level.

The database component is implemented using MySQL database as it is open source software and fits the size of the social data set used in the implementation. Hadoop [58], which is an open source implementation of MapReduce [59], is a different alternative for the Database Management System as mentioned in section 4.2.

The MapReduce technology is designed to run data-intensive applications that are distributed in nature. The reason behind not using this technology in this implementation is that the data set used is small enough in size in a way that it would not be able to take advantage of the great performance gains that the architecture offers. Instead, a design decision was made to use the MySQL database. Further implementations using MapReduce will take place in future research activities.

Among Hadoop, the big online communities that use it include Facebook [60] and Yahoo [61], which are able to take advantage of the performance gains that Hadoop offers to its users.

5.2.3. Traffic Situation Information

This component in the simulation of the socially aware transportation system is extremely important. Its importance stems from storing actual road and traffic situations in a database allowing other motorists (OSN users) to access this data and get up-to-date information on a real-time basis. Consequently, this component is responsible for storing all driving and road traffic conditions in a database.

In real life, this simulates users themselves reporting different situations either using their hand-held devices or onboard computers. The traffic situation database is connected to the OSN, ensuring that suggested alternatives are supported by the social trustworthiness of them. Refer to section 4.1 for more detail on the database design and implementation.

In order to measure driving speed and trip duration values, users' smartphones are used as traffic sensors. Most mobile platforms that are used these days have system calls that either measure these values or offer tools to facilitate the calculation of these values.

Among these platforms are:

- Android [62]

  getSpeed():a method to measure driving speed for users utilizing their Android operated phones while driving.

  Location():a method used to retrieve current location of the phone.

- iOS [63]

  speed: a property offered by the core location class *CLLocation* within iOS.

  startUpdatingLocation():a method to decide on a user's location through using the GPS service in their iPhones.

5.2.4. Real-Time Processing/Interaction classes

The two previously mentioned components form the two main parts of the simulation model. Unfortunately, there is no means of communication available for both of them to exchange information that will help simulate the purpose of the socially aware traffic system.

In order to achieve this interaction, classes had to be written to enable communication and information exchange to take place between both components. These classes were written in Python scripting language. The reason for using Python is that the TraCI component in SUMO is available and ready to work only with Python classes. Communication between both SUMO and the interaction classes takes place using a TCP client/server communication architecture that provides access to SUMO simulation from any external tool. According to this definition, SUMO acts as a server and the interaction classes act as the client.

5.3. Simulation Scenario

In this section, the simulation setup and parameters are discussed. Specific scenarios that were simulated are described to show how the model was implemented and used.

In the simulations, we have a road network, shown in Figure 5.1, where traffic usually flows from one point to another through a certain route that is predefined in the XML configuration files for SUMO. With each step, cars report their positions to the road network and their speed for each *timestep* of the simulation.

When a certain situation, an accident or slow traffic, is detected, vehicles report these situations to the main control node through roadside units that are simulated here by the main control class of the software framework. When any car enters the road at anytime in the simulation, it contacts the control software in order to check if there is any delay in traffic that may slow down the trip.

The control software checks the database for any reported situation that is recent to the departure time of the enquiring vehicle. If there is any situation reported in one of the road segments ahead of the travelling vehicle, a suggestion is usually given as an alternative from the reporting vehicle. This suggestion is supported by the social trustworthiness of it. The social trustworthiness of suggestions and reported data are calculated based on the social distance between both users whether it is first, second, or third degree, and the propagated trust is then calculated and introduced to the user asking for suggestions.

Trust calculation relies on many users if there is more than one motorist reporting the situation. In this case, an overall trust of all users has to be calculated and introduced to the enquiring user as an overall trust rating. Figure 5.2 shows a sample simulation run for this situation.

The enquiring vehicle operator has the choice to follow the suggestions supplied by the system or not based on the trust rating. In simulations, this is accomplished by making a comparison against a predetermined threshold that represents the minimum overall trust rating that should be accepted by the user.

5.4. Experimental Setup

This section will show different parameters used within the experiments. In this set of experiments, a number of 9 vehicles participate in the simulations; vehicles are identical in their characteristics, acceleration, deceleration, length, and maximum speed.

Using 2 different routes, one route represents the main route that vehicles would typically use and the other is the alternative route that will be suggested if there is blockage in the first one. The maximum speed for road segments is 50 km/h. The 9 different experiments were performed using the friends settings shown in Table 5.1. Each experiment was run between 7-9 times, and average results were calculated for each simulation setting. The range was chosen depending on when the results reached steady state and there was no longer significant change between simulation runs.

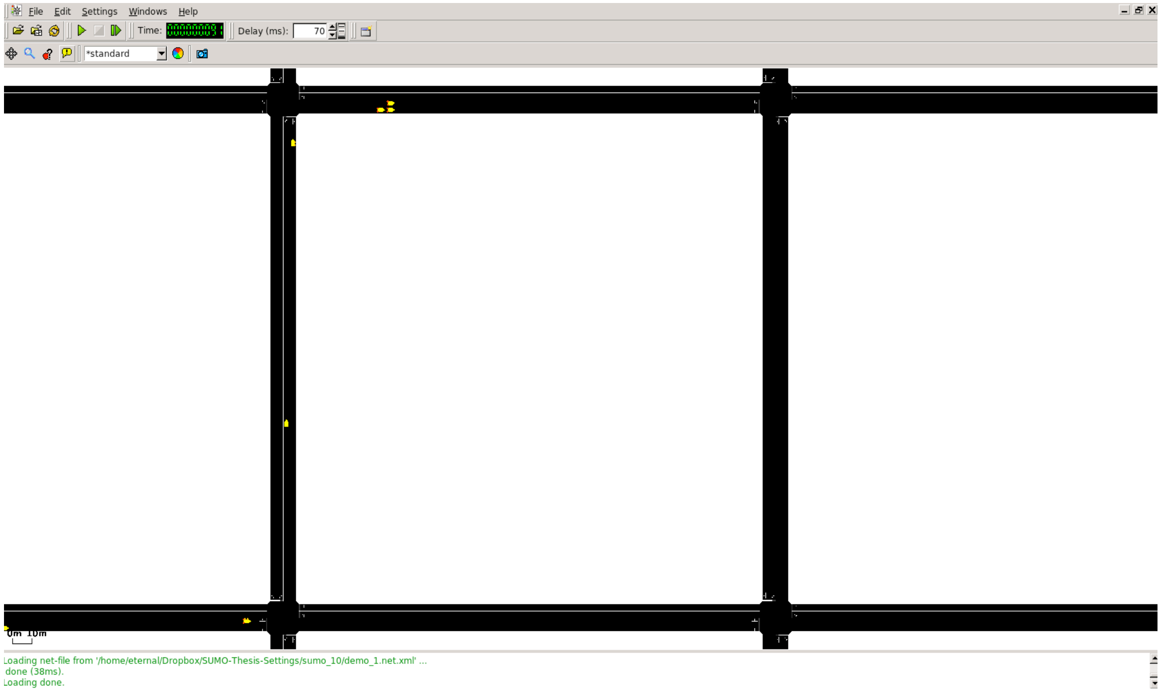

Figure 5.2: Sample Simulation Run

Exp no	1^{st}	2^{nd}	3^{rd}	# Users (Vehicles)
1	1	-	-	1
2	1	1	-	2
3	1	1	1	3
4	2	1	1	4
5	2	2	1	5
6	2	2	2	6
7	3	2	2	7
8	3	3	2	8
9	3	3	3	9

Table 5.1: Simulation Settings

Component	Name/Version
OS	Ubuntu 10.10
Kernel	Linux 2.6.35
CPU	Core 2 DUO 2.4GHz
RAM	6 GB
MySQL	5.1.49
Python	2.6.6

Table 5.2: Simulation Environment

Table 5.2 shows computer settings used to perform the experiments.

5.5. Data Sets

The OSN dataset used for the purpose of simulations was extracted from a peer-to-peer file-sharing gnutella [64] network. The main reasons for this choice are that it is:

- A free dataset that is available for the general public to use and experiment with social relationships within the network.
- A large enough dataset to reflect social relationships.
- A real-life representation of an online social network.

This social network contains 6301 unique nodes. The number of edges connecting pairs of nodes is 20,781, reflecting their social relationships. For the purpose of simulations, we needed to have Directed Trust Values (DTVs) for each pair of friends. Each value represents the trust of one user in the other within each pair. These trust values led to 41,562 directed trust values, representing all relationships in both directions. Figures 5.3 and 5.4 show a social graph for the social dataset.

Figure 5.3 shows a social graph that includes all the nodes and the edges of it. As can be seen, it is a very dense representation of the graph since it includes a large number of nodes and larger number of edges.

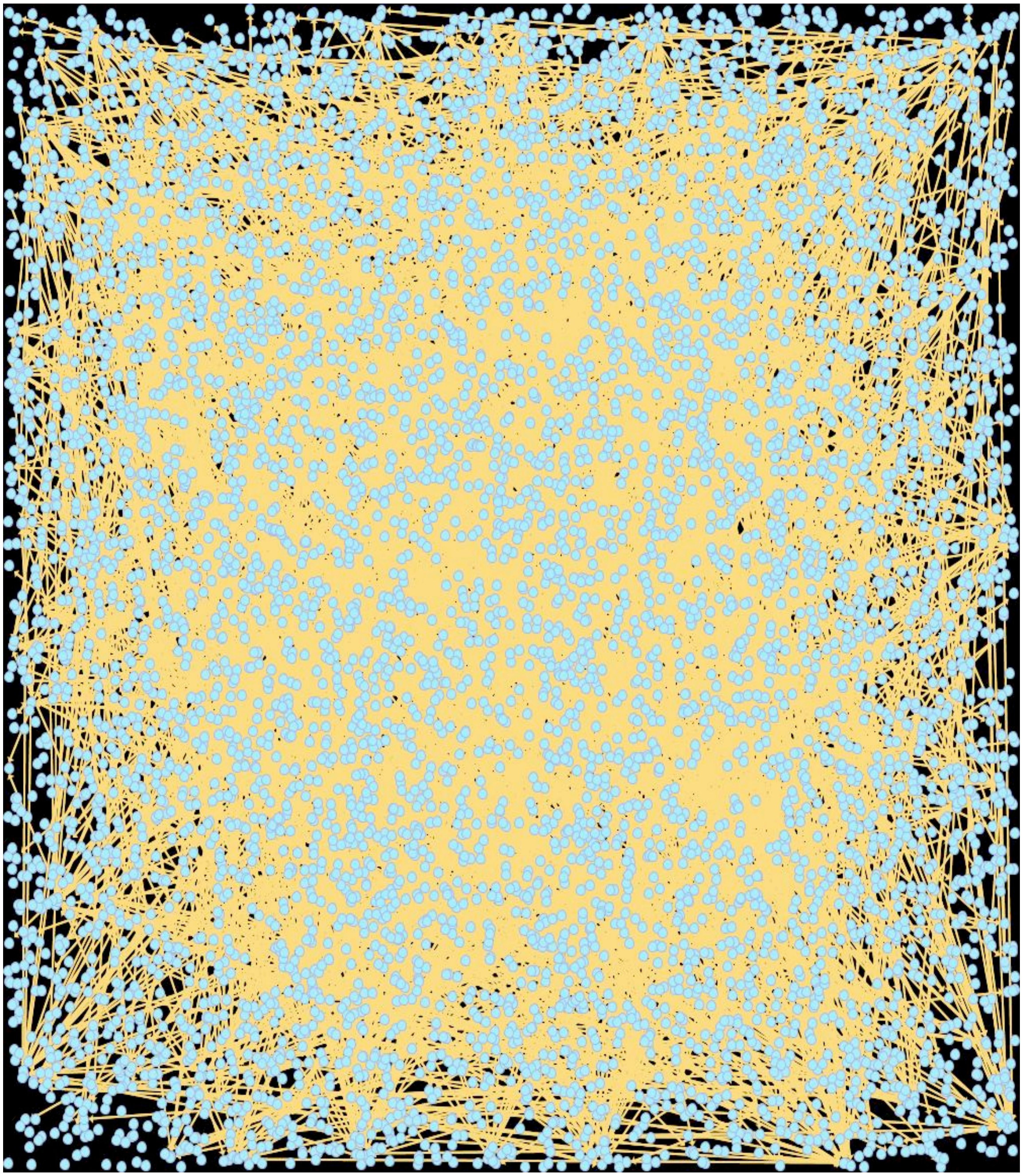

Figure 5.3: Social Graph

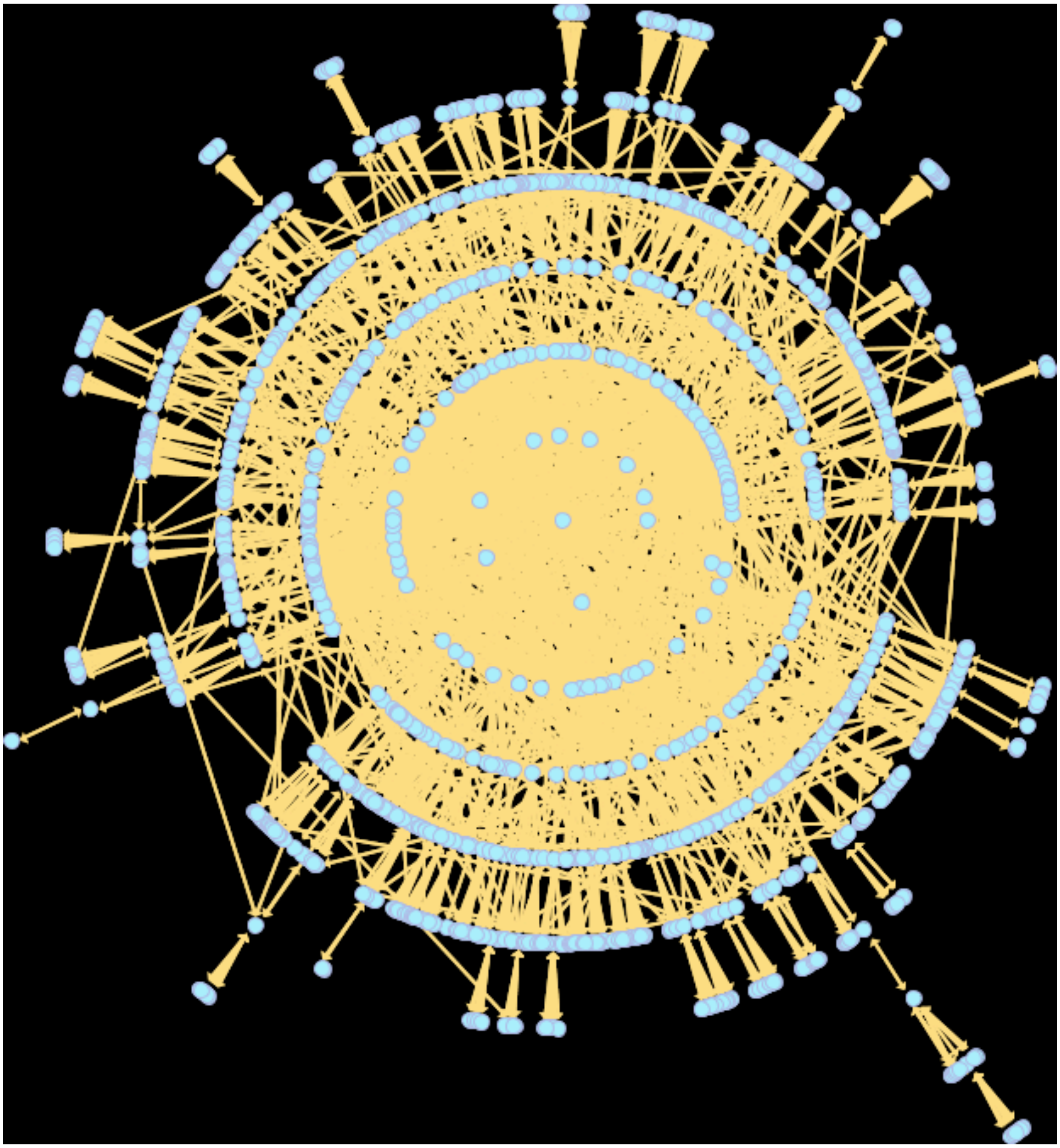

Figure 5.4: Circular Social Graph

Figure 5.4 is another representation of the social graph in which a circular representation is shown. All nodes in the system are centralized around a central node, which is node number 1 in this graph. Figure 5.4 shows clearly the different degrees of friendship between the central node and all other nodes in the system.

A random generation method within Python scripting language was used in order to generate DTVs between each pair of users to cover all trust relationships between individual pairs.

5.6. Simulations

Trust inference is being simulated in this section to show how trust propagation among nodes will affect trust values between nodes who are first degree friends (direct neighbours), have a second degree relationship (friends of friends), or a third degree relationship (friends of friends of friends). Trust inference in this case is affected by the degree of friendship between both *truster* and *trustee*. Also, the simulations show trust inferences and how they are affected by suggestions from many sources, some of which are first, second, and/or third degree relationships from the enquirer. The aggregation of these trust values, from multiple sources, affects the overall trust in recommendations, as well as the processing time to generate trust.

Also, the global trust ratings that each user collects from past recommendations made by them were utilized in another group of simulations. These simulations tested situations in which there is no social trust recommendation within the three degrees of friendship. That means *trustees* are network users but propagated social trust cannot be calculated for them. Therefore, another metric, the global trust in trustees, was used and tested in this simulation. *Trustees* obtain these global ratings whenever they make recommendations to others and get feedback about these recommendations, either positive or negative.

Simulations were made on different settings to explore different types of friendship options to show, as previously mentioned, how these affect cumulative trust. In each user setting, two main values were extracted; the propagated trust over different trust paths and the processing time to generate this propagated trust value.

For the global trust simulations, simulation software was used to simulate 9 experiments, as well, starting from 1 recommender to 9 of them. In this case, every experiment had to test the relationship strength first and, if no recommendation was coming from any of the three degrees of friendship, global trust method was invoked to decide whether to trust recommenders or not, and what level of trust is expressed and whether it is higher or lower than the pre-set threshold for that truster.

Another group of simulations was performed using a simple average as a means of calculating trust instead of those used in Equations 4.1, 4.2, and 4.3. So far, trust calculation has been tested on different degrees of friendship, and categorization within degrees has been made. This categorization tends to classify users within three groups: family members, friends, and co-workers. This categorization tends to add another social dimension to simulations to make it more realistic.

5.7. Simulation Scenarios

This section lists possible scenarios for applications that can be implemented with the proposed solution.

1- *City map* guided with global trust (reputation)

In this application, the use of a global trust rating similar to the one used within eBay is incorporated. Users feed speeds they are travelling at using their cell phones, and this data is then processed and their global trust values are analyzed. Based on the trust analysis, information from certain users only is chosen to be used as part of the overall recommendation system.

The intended application is to build a city map that can reflect speeds of road segments based on information fed by users. The map will cover areas that users travel frequently and feed information about the route through their trips.

A very important factor here that will define which information to use from users is their global trust rating. Users gain their global trust rating based on their history of participation and feedback from other users in the system. Two scales could be used in this case:

- A pointing system, in which users gain points each time they make recommendations to the system and they gain positive feedback about their recommendations. The higher the value for each user, the more trustworthy that user is.

- Global trust value could be in the form of a percentage; in this form, global trust is represented by the percentage of correct/useful recommendations of all recommendations a certain user has made.

In this type of application, the number of participants is expected to be within the range of thousands of users. The greater the number of participants, the more trustworthy the information will be.

2- *Freight Trust*

In this type of application, social network trust is used for commercial purposes. This application can be applied on two domains:

- Within the same company: As an example, a taxi company that covers a certain city can benefit from information about traffic conditions supplied by the company's drivers.

- Among companies: This domain is a broader one, wherein drivers from a certain freight company can make use of traffic information made available by another freight company's traffic information.

In the second case, there are two types of trust levels. One level covers drivers within the same company. The other level covers trust transitivity between drivers of two or more different companies. The coverage of this application falls within the range of hundreds of users depending on the number of participant companies and how many drivers they each have to feed traffic information.

3- *Social traffic calendar app*

This application merges two types of information, traffic and calendar information. Consider a situation in which a user is in a meeting and has a flight to catch later in the day. The traffic application can pull the user's calendar information and check road conditions to derive the Estimated Time of Arrival (ETA) to the airport. Based on calendar events, the application checks other users traffic feeds and estimates when the user should leave the meeting in order to be able to catch the flight.

This type of application makes use of information supplied by other travelers on the same flight who are coming from different places within the same city to derive the ETA. Based on those users' trust ratings, alternative routes and estimated time decisions can be guided with derived social trust ratings. The number of users that might be participating on such applications may fall in the range of 10s or maybe 100s of users depending on a given event's number of participants.

4- *Traffic Trust*

This application is a general case scenario in which a user is about to travel between certain locations and needs to get recommendation information about traffic conditions. The user consults the app for that, and the application searches the user's social network for any person (family, co-worker, friend) who gave recommendations. Recommendation information is then analyzed. Traffic decisions are made and supplied to the user with social trust ratings on how trustworthy this information is.

5.8. Results

This section shows results obtained from simulations of different situations wherein different driving conditions were applied to show the system's response, behaviour, and performance for various traffic and demand conditions. Overall trust score is amongst the important values in simulations and will be drawn against the number of users participating in forming suggestions in each case. Processing time required to infer the degree of friendship and propagated trust is important, as well, to show how well the system is performing.

Trust distribution for all participants in simulations that are expressed by the truster are used first to show how this graph forms the basis of the calculations for propagated trust inference. Figure 5.5 shows a trust distribution of the trust each *truster* has in each *trustee* individually.

Values for trust rating in each degree of friendship varies between experiments depending on the number of participants and their DTVs. Accordingly, Figure 5.6 shows trust the distribution over each degree of friendship for each experiment conducted with the settings mentioned in Table 5.1.

Trust ratings are propagated values for the truster from each and every *trustee* participating in forming a recommendation. Propagated trust in this case is a simple average depending on the degree of friendship, which has to be with the three degrees of relationship (first, second, or third). Values in Figure 5.5 were used for simulations that examined local trust ratings, while those in the global trust experiments are not shown as they are outside of the three degrees of friendship.

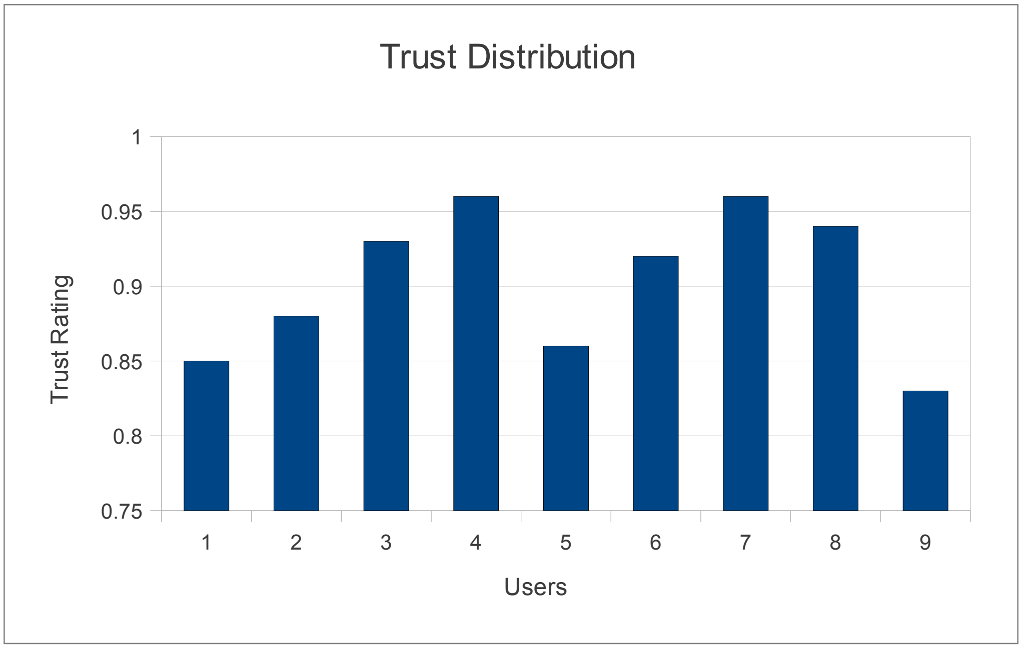

Figure 5.5: Trust Rating of the Truster in Each Trustee

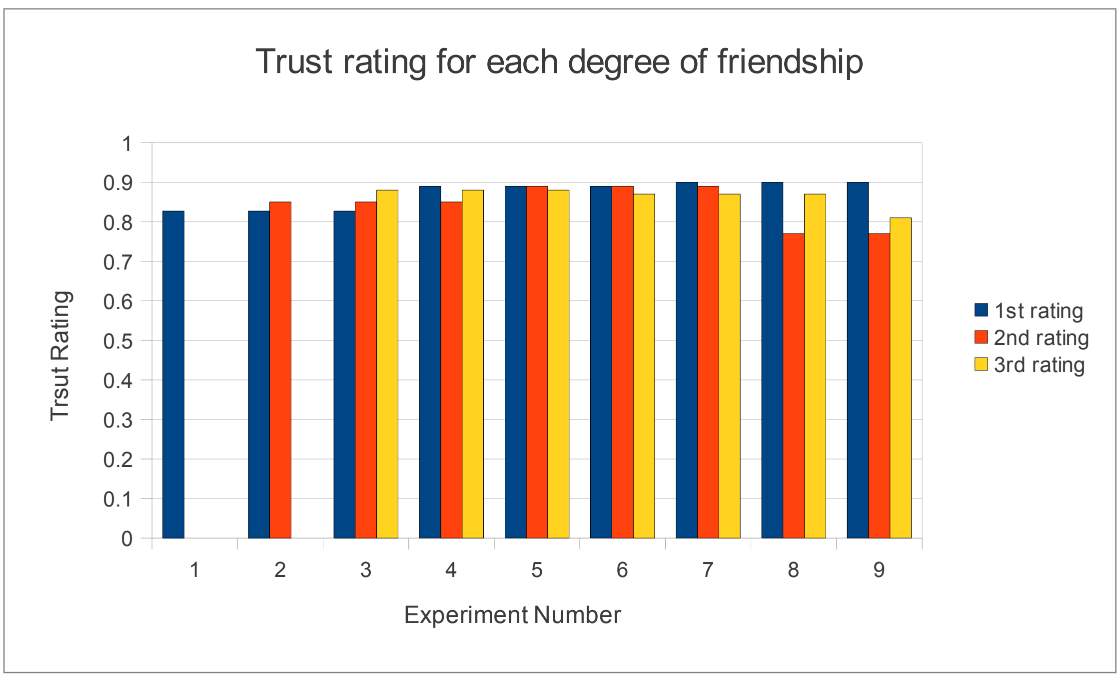

Figure 5.6: Trust Ratings for Different Degrees of Friendship

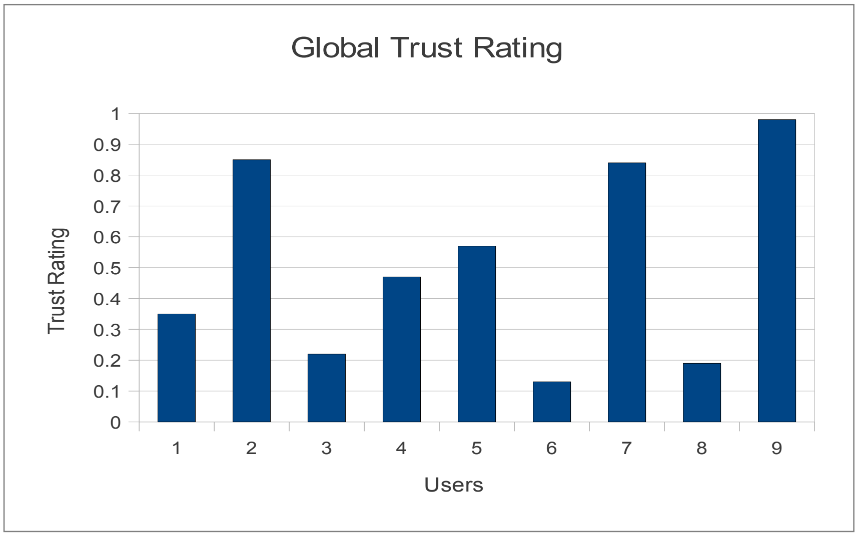

Figure 5.7: Global Trust Ratings for Trustees

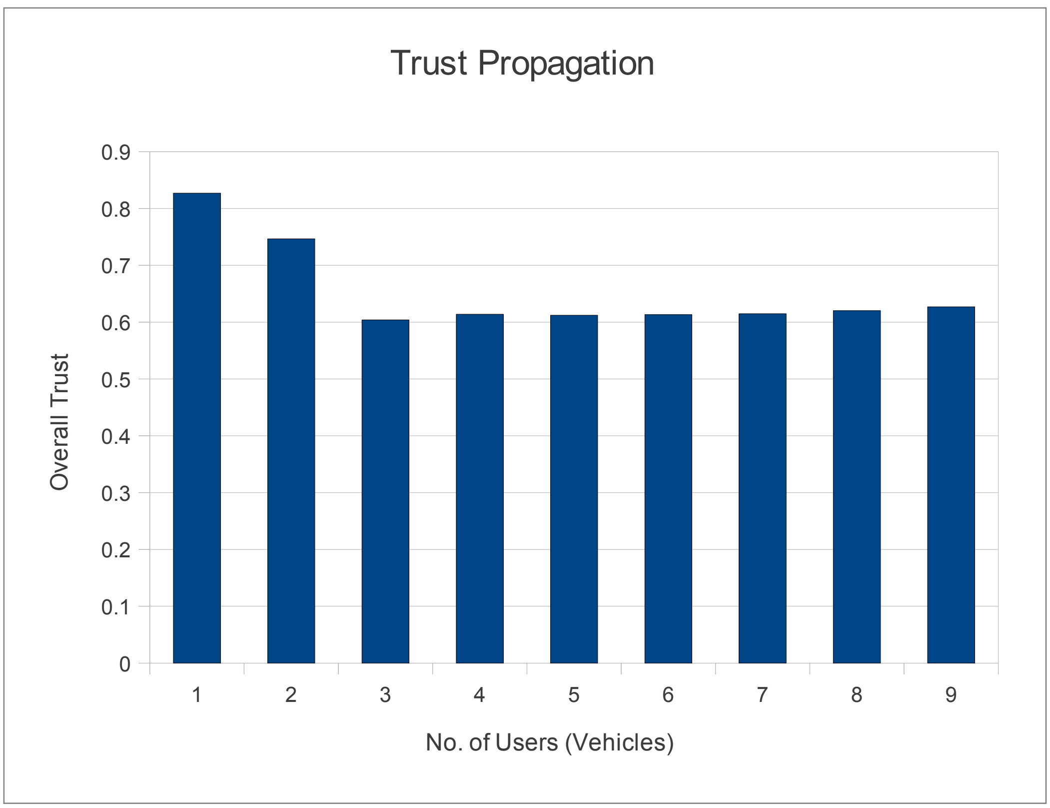

Figure 5.8: Overall Trust for all Participants in Each Experiment

Instead, Figure 5.7 shows global trust ratings for users participating in the global trust experiments. These values are fixed, which means if any recommendation is coming from these users while they are outside of the three degrees of friendships of the truster, these global values are given no matter who the truster is.

The groups of users in Figures 5.5 and 5.7 are not the same. The first group has social proximity that we can use to calculate propagated social trust and then use them for overall trust ratings while the users in the second group have no social proximity that can be utilized so we use their global trust ratings.

Given the data in Figure 5.7, trust calculation took place utilizing these values using Equations 4.1, 4.2, and 4.3 to infer propagated trust the truster has in the trustees. Figure 5.8 shows the overall trust propagated in each simulation setting of the nine scenarios mentioned earlier.

As a comparison of propagated trust values in both cases of weighted trust and simple average trust propagation, Figure 5.9 shows the differences in each case of the 9 experiments. We can infer from Figure 5.9 that the results of weighted trust are more accurate than those of the simple average ones.

This conclusion comes from the assumption that all participants from all closeness levels should not participate in the trust equation with the same share; instead, those who are closer in relationship should participate with higher values of trust than those who are more distant.

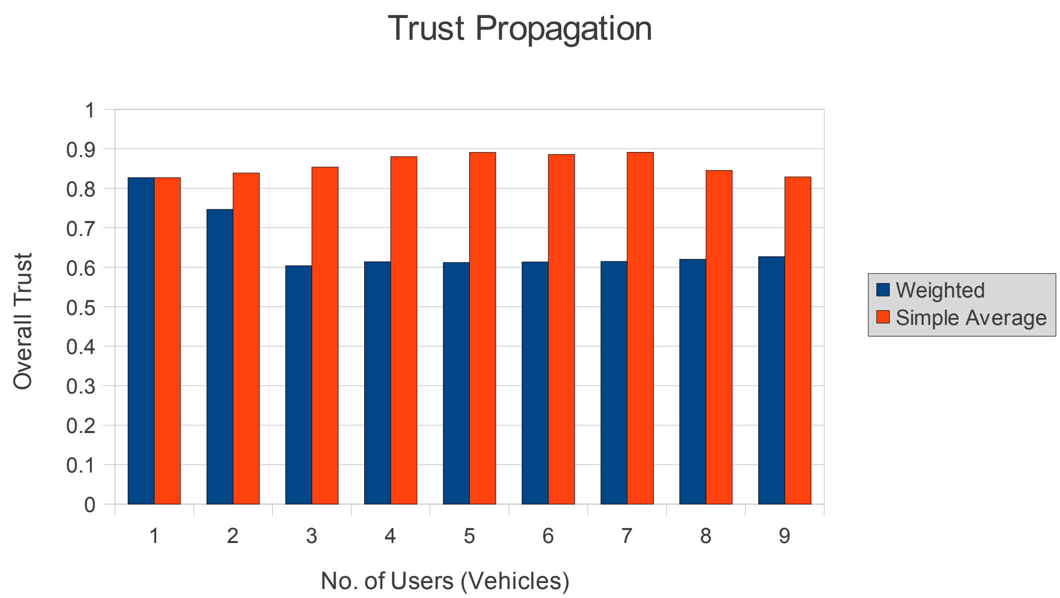

Figure 5.9: Trust Propagation in Weighted Vs. Simple Average Cases

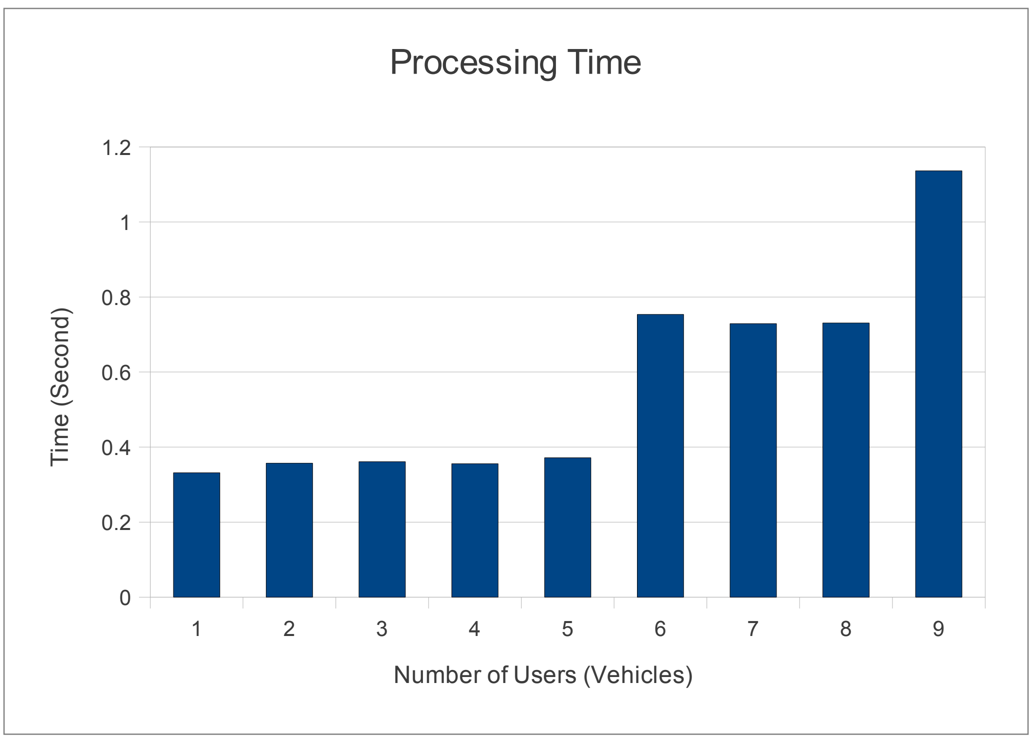

Figure 5.10: Processing Time for the Overall trust Calculation

Processing time is the main cost for obtaining the values in Figure 5.8. Processing time is actually divided into two main values. The first is the walk time, and it is defined as the time required for the algorithm to walk through the relationship network and decide which degree of friendship both the *truster* and the *trustee* incorporate. The second value is the calculation time, and this value is defined as the time required for calculating the overall trust rating for all *trustees* after obtaining the degree of relationship between users and each DTV between individual pairs of them.

Figure 5.10 shows the processing time for the overall trust including both the walk time and the trust calculation time.

Figure 5.11 shows the time difference in trust propagation calculation in the case of weighted trust propagation versus the simple average one. This figure shows that the difference in processing time as a cost of trust calculation in both cases is slightly small. It also shows that weighted trust usually takes more time to calculate, except in two cases in which simple average time is a bit higher than the weighted one. Normally, calculations in the weighted trust case are a bit more time consuming than the simple average that resulted in these situations.

Figure 5.12 shows the processing time required for global trust generation in the cases in which users do not fall within the three degrees of friendship.

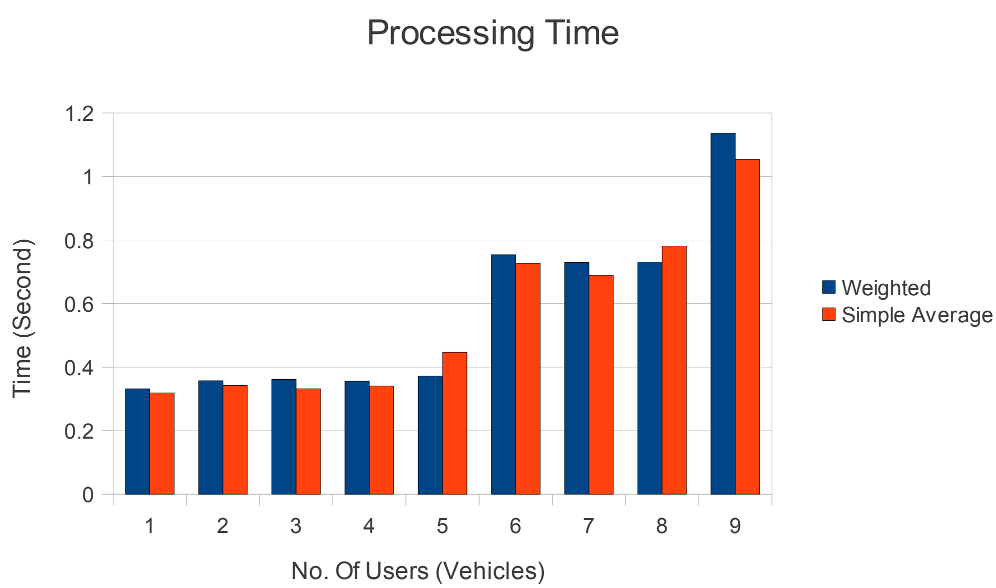

Figure 5.11: Processing Time for the Overall Trust Calculation Both in Weighted and Simple Average Cases

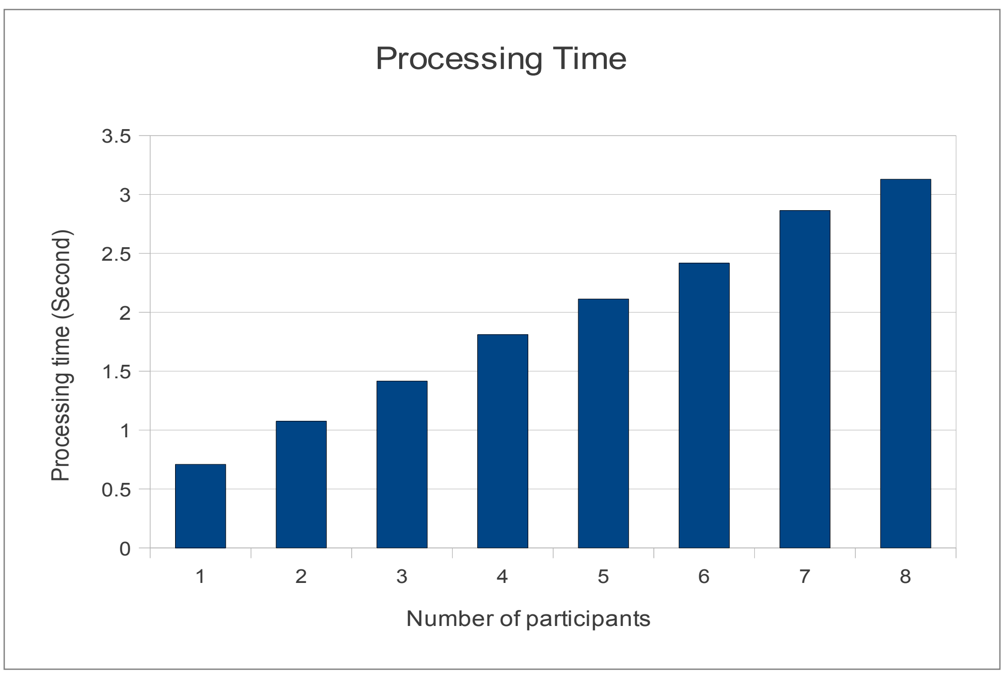

Figure 5.12: Processing Time for the Overall Trust Calculation in Case of Global Trust Rating

SUMMARY

In this chapter, a simulation model was designed in order to prove the concept of socially aware ITS applications and analyze solution performance as well. All components of the simulation system were explained along with how they work together. Data sets used for the purpose of simulation were introduced. Simulation scenarios for possible ITS applications were explained. Results were shown, and analyses of results were made in order to explain and analyze system behaviour under different simulation conditions.

6. CONCLUSION

This work introduced the idea for socially supported recommendations in transportation systems, especially in the proposed 4 novel applications. Within this scope, the concept of trust calculation in socially aware traffic was discussed. Directed social trust between individual users was utilized to express how confident a certain user is in a friend's recommendations. Trust calculation was extended to cover deeper degrees of friendship beyond first-degree relationships. Second and third degree relationships were used by inferring propagated trust between individual non-friend users, taking advantage of all the relationships in between.

Global trust ratings are used as well to represent users' reputations in the system in case a recommender does not fall within the three degrees of friendship with the user asking for a recommendation.

Traffic information supplied to the system is analyzed, and alternative routes are extracted. Trust propagation between *trusters* and *trustees* in the social network is calculated using an empirical equation to make use of social connections among system users. Social trust in recommenders is then calculated. Guided by derived social trust, traffic alternatives are supplied to users on their mobile devices, or vehicle on-board computer

Both trust ratings based on the proposed empirical equations and the simple average techniques were measured with their processing times to show the overhead added in using those empirically propagated trust generation equations.

Empirical trust equations are more reliable and expressive for propagated trust as they take into consideration the degree of friendship between users. Different weight values were used in the trust propagation equation to show how much these weights affect the results. Also, different user categories (Friends, Family, Co-works) were used to show how results could be affected by incorporating different user groups with different trust weights.

A complete architecture for a software solution that merges social recommendations with intelligent transportation systems and supporting those recommendations with social trust between individual users was shown. Such a system can help reduce traffic congestion in urban areas using social networks and Internet-capable devices that increase the number of users.

Simulation scenarios were created to discuss the different aspects and conditions that might be encountered in the software application. Also a simulation platform was created to test these scenarios using open source software components. These simulations show how applicable and efficient such a system would be in a world of increased usage of the Internet, especially with portable devices, as well as increased traffic density in urban areas.

In conclusion, the inclusion of social trust derived from OSNs is valuable in increasing confidence in shared information in ITS applications. It also helped propose a set of new ITS applications, that offers new features to users, supported by social trust in recommending users.

Further research and development is required to enhance this concept:

1. Before making traffic suggestions available to other users in the social network, there should be a minimum number of users reporting a certain situation, as this will increase system credibility among users.

2. Since this system is intended to be a real-time application, information criticality and value should change with time especially if no further reports are coming from users of the application about certain situations that happened in the past.

3. The integration of this application with V2V applications needs to be investigated as this integration could enhance performance and credibility of V2V applications on the road.

REFERENCES

[1] Thomas F. Golob, Amelia C. Regan, Impacts of information technology on personal travel and commercial vehicle operations: research challenges and opportunities, Transportation Research Part C: Emerging Technologies, Volume 9, Issue 2, April 2001, Pages 87-121

[2] J.R. Wootton, A. García-Ortiz, S.M. Amin, Intelligent transportation systems: A global perspective, Mathematical and Computer Modelling, Volume 22, Issues 4–7, August–October 1995, Pages 259-268

[3] Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. 2007. Measurement and analysis of online social networks. In Proceedings of the 7th ACM SIGCOMM conference on Internet measurement (IMC '07). ACM, New York, NY, USA, 29-42

[4] Huang, G., "Powerful smartphone solutions unleashing new technology innovations," VLSI Design, Automation, and Test (VLSI-DAT), 2012 International Symposium on, vol., no., pp.1,1, 23-25 April 2012

[5] Yanjun Zuo, Wen-Chen Hu, Timothy O'Keefe; "Trust Computing for Social Networking".

[6] Guanfeng Liu, Yan Wang, Mehmet Orgun; "Trust Inference in Complex Trust-oriented Social Networks".

[7] R. Miller, D. Perlman, and S. Brehm. Intimate Relationships. McGraw-Hill College, 4th edition, 2007.

[8] R. Ashri, S. Ramchurn, J. Sabater, M. Luck, and N. Jennings. Trust evaluation through relationship analysis. In Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'05), pages 1005– 1011, Netherlands, 2005.

[9] Y. Wang and V. Varadharajan. Role-based recommendation and trust evaluation. In Proceedings of the IEEE Joint Conference on E-Commerce Technology and Enterprise Computing, E-Commerce and E-Service, pages 278–295, Tokyo, Japan, July 2007.

[10] J. Heer and D. Boyd. Vizster: Visualizing online social networks. In Proceedings of the 2005 IEEE Symposium on Information Visualization (InfoVis'05), pages 5–12, USA, October 2005.

[11] C. Lin, N. Cao, S. Liu, S. Papadimitriou, J. Sun, and X. Yan. Smallblue: Social network analysis for expertise search and collective intelligence. In Proceedings of the 2009 IEEE International Conference on Data Engineering (ICDE'09), pages 1483–1486, Shanghai, China, March 2009.

[12] E. Bender. Bayesian network. Mathematical Methods in Artificial Intelligence, pages 301–311, 1996.

[13] Katarina Stanoevska-Slabeva, Katarina Stanoevska-Slabeva, Thomas Wozniak, Isabella Hoffend, Jana Ebermann; Towards a Concept for Inclusion of Social Network Information as Context Information.

[14] Boyd, D. and Ellison, N. (2007). Social network sites: Definition, history, and scholarship. Journal of Computer-Mediated Communication, 13(1):210-230.

[15] Ellison, N. B., Steinfield, C., and Lampe, C. (2007). The benefits of facebook "friends:" social capital and college students use of online social network sites. Journal of Computer-Mediated Communication, 12(4):1143-1168.

[16] Choi, J. H. (2006). Living in Cyworld: Contextualising Cy-Ties in South Korea. In A. Bruns, J. Jacobs (Eds.): Use of Blogs (Digital Formations), pp. 173-186. New York: Peter Lang.

[17] Schilit, B., Adams, N. Want, R. Context-Aware Computing Applications. 1st International Workshop on Mobile Computing Systems and Applications. (1994) 85-90.

[18] Dey A. K. & Abowd, G. D. (2000). Towards a better understanding of context and contextawareness. Proceedings of the Workshop on the What, Who, Where, When and How of Context-Awareness, affiliated with the CHI 2000 Conference on Human Factors in Computer Systems, New York, NY: ACM Press.

[19] Zimmermann, A. (2007). Context Management and Personalisation: A Tool Suite for Context- and User-Aware Computing. Dissertation thesis,RWTH Aachen.

[20] Eisend, M (2006) "Source Credibility Dimensions in Marketing Communication - A Generalized Solution", Journal of Empirical Generalisations in Marketing Science, Vol 10, No.2.

[21] Brown, J., Broderick, A. J., and Lee, N. (2007). Word of mouth communication within online communities: Conceptualizing the online social network. Journal of Interactive Marketing, 21(3):2-20.

[22] Mohsen Jamali, Martin Ester; TrustWalker: A Random Walk Model for Combining Trust-based and Item-based Recommendation.

[23] Paolo Avesani, Paolo Massa, Roberto Tiella; A Trustenhanced Recommender System application: Moleskiing.

[24] P. Massa, P. Avesani; Trust-aware collaborative filtering for recommender systems. In Proc. of Federated Int. Conference On The Move to Meaningful Internet: CoopIS, DOA, ODBASE. Springer-Verlag LNCS 3291, 2004.

[25] C. Ziegler and G. Lausen. Spreading activation models for trust propagation. In IEEE International Conference on e-Technology, e-Commerce, and e-Service (EEE'04), 2004.

[26] Jennifer Golbeck, James Hendler, and Bijan Parsia. Trust networks on the Semantic Web. In Proceedings of Cooperative Intelligent Agents, 2003.

[27] Carminati, B., Ferrari, E. and Perego, A. (2006). Rule-based Access Control for Social Networks. Proceedings of the IFIP WG 2.12 and 2.14 Semantic Web Workshop, November, Montpellier, France.

[28] Jennifer Golbeck; Combining Provenance with Trust in Social Networks for Semantic Web Content Filtering.

[29] J. Golbeck. Computing and Applying Trust in Web-based Social Networks. Ph.D. Dissertation, University of Maryland, College Park, 2005.

[30] J. Golbeck. Generating Predictive Movie Recommendations from Trust in Social Networks. Proceedings of The Fourth International Conference on Trust Management, 2006.

[31] D. Brickley and L. Miller. Foaf vocabulary specification. http://xmlns.com/ foaf/0.1/, 2005.

[32] I. Davis and E. V. Jr. Relationship: A vocabulary for describing relationships between people. 2004.

[33] Ee-Peng Lim, Aixin Sun, Jaideep Srivastava, Young Ae Kim; Predicting Trusts among Users of Online Communities – an Epinions Case Study.

[34] http://www.epinions.com.

[35] Cai-Nicolas Ziegler, Georg Lausen; Analyzing Correlation between Trust and User Similarity in Online Communities

[36] Golbeck, J. (2006). Trust and nuanced profile similarity in online social networks

[37] Dijiang Huang; Zhibin Zhou; Xiaoyan Hong; Gerla, M.; "Establishing Email-Based Social Network Trust for Vehicular Networks," *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE* , vol., no., pp.1-5, 9-12 Jan. 2010

[38] Paolo Massa, Paolo Avesani; "Trust-aware recommender systems", Proceedings of the 2007 ACM conference on Recommender systems, October 19-20, 2007, Minneapolis, MN, USA

[39] CHEN Qun; Dynamic Route Guidance Method Based on Particle Swarm Optimization Algorithm. 2009 Second International Conference on Intelligent Computation Technology and Automation.

[40] D. Schrank and T. Lomax, "The 2007 Urban Mobility Report", Texas Transportation Institute, The Texas A&M University System, 2007.

[41] Dijkstra E W, "A note on two problems in connection with graphs", Numerische Mathematic, 1959, 1 (5): 269-271.

[42] Floyd R N. Algorithm 97 shortest path [J].Comm. ACM, 1962, 5(6):345.

[43] A Crauser, Kmehlhorn, Umeyer, Psanders. "A parallelization of

Dijkstra's shortest path algorithm", Brno, Czech Republic, Springer. Proceedings of the 23rd Symposium on Mathematical Foundations of Computer Science. Berlin: Lecture Notes in Computer Science, 1998.4

[44] Anjali Awasthi, Yves Lechevallier, Michel Parent and Jean-Marie Proth; Rule based prediction of fastest paths on urban networks

[45] Wang Hongde, Cui Tiejun, Wang Shiyu; Traffic Route Dynamic Guidance Based on Coupling of Time Recursive and Artificial Neuron Network

[46] Kai Chen, Kia Makki, and Niki Pissinou; A Real-Time Wireless Route Guidance System for Urban Traffic Management and Its Performance Evaluation.

[47] V. N. Vapnik, "The Nature of Statistical Learning Theory", Springer, 1995.

[48] V. N. Vapnik, "An overview of statistical learning theory", IEEE Trans. Neural Networks, vol. 10 pp. 988-999,Sept. 1999.

[49] C. H. Wu, J. M. Ho, and D.T. Lee, "Travel-Time Prediction With Support Vector Regression", IEEE Trans. Intelligent Transportation Systems, Vol. 5 No. 4, 2004.

[50] R. Chrobok, O. Kaumann, J. Wahle, and M. Schreckenberg, "Three categories of traffic data: Historical, current, and predictive", in Proc. 9th Int. Fed. Automatic Control Symp. Control in Transpotation Systems, 2000, pp. 250-255.

[51] J.-W. Ding, C.-F. Wang, F.-H. Meng, T.-Y. Wu; "Real-time vehicle route guidance using vehicle-to-vehicle communication"

[52] K. Park, M. Bell, I. Kaparias and K. Bogenberger; Learning user preferences of route choice behaviour for adaptive route guidance.

[53] Quinlan, R.: 'C4.5: Programs for machine learning' (Morgan Kaufmann Publishers, Inc., 1993)

[54] Matthias Gerlach; Stephen Steglich; Stefan Arbanowski; Maarten Wegdam; Harold Teunissen; "Trustworthy applications for Vehicular Environments,"*Vehicular Technology Magazine, IEEE*, vol.1, no.2, pp.9-15, 2006.

[55] Gerlach, Matthias; "Trust for Vehicular Applications,"*Autonomous Decentralized Systems, 2007. ISADS '07. Eighth International Symposium on*, vol., no., pp.295-304, 21-23 March 2007

[56] Shuo Ma, Ouri Wolfson, and Jie Lin; "A survey on trust management for intelligent transportation system". In Proceedings of the 4th ACM SIGSPATIAL International Workshop on Computational Transportation Science (CTS '11). ACM, New York, NY, USA, 18-23, 2011.

[57] http://sumo.sourceforge.net/

[58] http://hadoop.apache.org

[59] https://developers.google.com/appengine/docs/python/dataprocessing/overview

[60] http://hadoopblog.blogspot.ca/2010/05/facebook-has-worlds-largest-hadoop.html

[61] http://developer.yahoo.com/hadoop/

[62] http://developer.android.com

[63] http://developer.apple.com

[64] http://snap.stanford.edu/data/p2p-Gnutella08.html