A WEB SERVICE BASED TRUST AND REPUTATION SYSTEM FOR TRANSITORY COLLABORATION FORMATION IN SUPPLY CHAINS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

İBRAHİM TAŞYURT

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

JULY 2009

Approval of the thesis:

A WEB SERVICE BASED TRUST AND REPUTATION SYSTEM FOR TRANSITORY COLLABORATION FORMATION IN SUPPLY CHAINS

submitted by **İBRAHİM TAŞYURT** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Müslim Bozyiğit _____
Head of Department, **Computer Engineering**

Prof. Dr. Asuman Doğaç _____
Supervisor, **Computer Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. İ. Hakkı Toroslu _____
Computer Engineering Dept., METU

Prof. Dr. Asuman Doğaç _____
Computer Engineering Dept., METU

Prof. Dr. Özgür Ulusoy _____
Computer Science Dept., Bilkent University

Assoc. Prof. Dr.Ahmet Coşar _____
Computer Engineering Dept., METU

Dr. Gökçe Banu Laleci Ertürkmen _____
Deputy Manager, SRDC Ltd.

Date: 10.07.2009

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: İbrahim Taşyurt

Signature :

ABSTRACT

A WEB SERVICE BASED TRUST AND REPUTATION SYSTEM FOR TRANSITORY COLLABORATION FORMATION IN SUPPLY CHAINS

Taşyurt, İbrahim

M.S., Department of Computer Engineering

Supervisor: Prof. Dr. Asuman Doğaç

July 2009, 74 pages

Today, advancements in the information technologies increased the significance of electronic business in the world. Besides the numerous advantages provided by these advancements, competition has also increased for the enterprises. In this competitive environment, companies have to access information faster and response to the changes quickly.

In a supply chain, it is a highly possible that one of the partners may defect in providing its services. When these exceptional cases occur, the pending parties have to establish transitory collaborations to replace the missing partner promptly in order not to suffer this deficiency economically. Companies need to know the competences and capabilities of their prospective business partners before establishing partnerships. Furthermore, the reputations of the candidate partners have to be known to avoid possible regrettable partnerships.

In this thesis, we have developed a trust and reputation model that can be used over supply chains to determine and exploit the reputation of providers in transitory collaboration formation. The trust model takes the behaviors of providers,

consumers into account and combines multiple criteria to aggregate a single reputation value. Experimental results show that, our model provides a robust and reliable reputation mechanism addressing a number of issues that have not been covered in the related studies.

In addition to this, an implementation of the model is realized within a Web application and the functionalities have been exposed as Web Services. The interoperability of the Web Services have been ensured through standard GS1 XML documents, which are utilized and extended in scope of the thesis. Furthermore, client interaction is provided through Web based user interfaces and REST services.

Keywords: Trust, Reputation, GS1 XML, Supply Chain Management, Web Services

ÖZ

TEDARİK ZİNCİRLERİNDE GEÇİCİ İŞBİRLİKLERİ KURULABİLMESİ İÇİN WEB SERVİS TABANLI GÜVEN VE İTİBAR SİSTEMİ

Taşyurt, İbrahim
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Asuman Doğaç

Temmuz 2009, 74 sayfa

Günümüzde, bilgi teknolojilerindeki gelişmeler elektronik ticaretin anlamını arttırmıştır. Bu gelişmelerin sağladığı sayısız avantajın yanında, şirketler için rekabet de yükselmiştir. Bu rekabetçi ortamda, şirketler bilgiye daha hızlı ulaşmak ve değişikliklere ivedi olarak cevap vermek zorundadır.

Bir tedarik zincirinde, ortaklardan birinin işlevini yerine getirememesi, hayli olası bir durumdur. Bu tip olağandışı durumlar gerçekleştiğinde işlevini yerine getirmeyen ortak nedeniyle etkilenecek paydaşlar, eksik ortağın yerini doldurabilmek için ivedi olarak geçici işbirlikleri kurmalıdır. Şirketler ortaklık kurmadan önce, aday şirketlerin yeterlik ve yeteneklerini bilmelidirler. Buna ek olarak, olası pişmanlık verici ortaklıklardan kaçınmak için, ortak adaylarının itibarları da bilinmelidir.

Bu tez çalışmasında, tedarik zincirlerinde geçici işbirliklerinin kurulması için servis sağlayıcılarının itibaının belirlenmesi ve kullanılması amacıyla bir güven ve itibar modeli geliştirilmiştir. Bu güven modeli, sağlayıcıların ve kullanıcıların

davranışlarını göz önüne almakta, çoklu güven kriterlerinin birleştirerek tek bir bütünleşik itibar değeri oluşturmaktadır. Deneysel sonuçlar, modelimizin önceki çalışmalarda değinilmeyen konuları kapsayan, etkili ve güvenilir bir itibar mekanizması ortaya koyduğunu göstermektedir.

Buna ek olarak, söz konusu modelin gerçekleştirimi bir Web uygulaması içerisinde hayata geçirilmiş olup, işlevleri Web Servisler olarak kullanıma açılmıştır. Web Servislerin birlikte işlerliği, tez kapsamında kullanılan ve genişletilen standard GS1 XML dökümanları ile gerçekleştirilmiştir. Ayrıca, Web tabanlı kullanıcı arayüzleri ve REST servisleri ile kullanıcı etkileşimi de sağlanmıştır.

Keywords: Güven, İtibar, GS1 XML, Tedarik Zincirleri Yönetimi, Web Servisleri

To my family

ACKNOWLEDGEMENTS

First, I would like to state my sincere gratitude to my supervisor Prof. Dr. Asuman Doğaç for her understanding, encouragement and support throughout my graduate study.

I would like express my special thanks to Dr. Gökçe Banu Laleci Ertürkmen, for her enormous guidance, support and patience. Words are not enough to describe my gratitude and indebtedness.

I would also like to convey thanks to İ.Hakkı Toroslu and other jury members for their valuable comments on this thesis.

I am indebted to my past colleagues, forever friends Mustafa Yüksel and Mehmet Olduz, for their support during this study. I would also like to state my thanks to the SRDC members, especially Fulya Tunçer, Ali Anıl Sınacı and Güneş Aluç.

I would like to thank my colleagues in Milsoft ICT, for their patience and understanding.

I would also thank the Scientific and Technological Research Council of Turkey (TÜBİTAK) for providing the financial means during my graduate education.

I am deeply grateful to my father Abdullah Taşyurt, my mother Fügen Taşyurt and my brother Mehmet Ali Taşyurt. Without them, this work could not have been completed.

Finally, I would like to thank my very best friends: Alperen Toprak, Barış Yanar, Yücel Özkara and Kaan Meralan for their enduring patience and support that they provided whenever I needed.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ANN	Artificial Neural Networks
CPFR	Collaborative Planning, Forecasting, and Replenishment
DAO	Data Access Object
FP7	7th Framework Programme
GDSN	Global Data Synchronisation Network
GDSSU	Global Data Synchronization Service Utility
GLN	Global Location Number
GPC	Global Product Classification
GS1	Global Standard One
GTIN	Global Trade Item Number
HTTP	Hyper Text Transfer Protocol
iSURF	An Interoperability Service Utility for Collaborative Supply Chain Planning across Multiple Domains Supported by RFID Devices

JAXB	Java Architecture for XML Binding
JAX-WS	Java API for XML Web Services
JPA	Java Persistence API
JSON	JavaScript Object Notation
ORM	Object-Relational Mapping
PSM	Probability Similarity Match
REST	Representational State Transfer
SBD	Standard Business Document
SOAP	Simple Object Access Protocol
TCSU	Transitory Collaboration Service Utility
UBL	Universal Business Language
WSDL	Web Service Description Language
XML	eXtensible Markup Language
XSD	XML Schema Documents

CHAPTER 1

INTRODUCTION

The advancements in information technologies have leveraged the significance and importance of electronic business. Today, companies can electronically locate their customers and suppliers, interact with them and form partnerships. In addition to these, companies manage their internal processes and resources along with their supply chain by means of information technologies.

The advancements in the technology also increased the competition in the business. In order to survive in today's information powered, competitive and demanding world of business, companies need to access information faster; be more agile and responsive to the changes in the world. These necessities cannot be viewed as the concerns of individual companies only. From supplier to the retailer, the entire stakeholder parties in the supply chain need to collaborate in order to benefit from their business mutually.

This necessitated collaboration brings a number of requirements that have to be fulfilled by the stakeholders. First of all, the trading entities have to synchronize their data along the supply chain to access accurate information on time. In order to achieve this synchronization and realize all kinds of communication; business partners have to be interoperable seamlessly, which means that they need to use a common language in their information exchange. Furthermore, the partners need to align their planning processes to take the advantage of information technologies to a more extent.

In addition to this, exceptional events may occur, such as the deficiency of a supplier to provide supplies for their business partners; lack of quality; or latency in delivery. In these situations, companies need to update their plans and replace their providers

with suitable ones. This process has to be fast, as well as reliable for companies in order not to suffer from these exceptions.

iSURF Project [1], supported by European Commission Information Communication Technologies (ICT) Seventh Framework Programme (FP7) [2], aims to tackle the requirements stated above, in order to foster the involvement of companies(especially Small and Medium Enterprises) in collaborative planning, interoperable data exchange and data synchronization. This thesis work has been conducted as a part of iSURF Project. Details related to iSURF have been revealed in Section 2.2.

There are a number of initiatives targeting the aforementioned requirements. GS1 [2] is a global organization that defines the standards to improve the efficiency and visibility of supply and demand chains globally and across sectors. GS1 defines a number of standards to realize identification, content level interoperability, global data synchronization and combination of RFID technology in IT infrastructure. UBL (Universal Business Language) [4], which is a library of electronic business XML documents developed by OASIS [5], provides a set of business document formats to be exchanged among partners from different industries. For the alignment of planning processes, CPFR (Collaborative Planning, Forecasting and Replenishment) [6] is a concept aiming to enhance supply chain integration, assisting joint processes and planning.

The exceptional cases that may lead to defectiveness of a partner have a high-level likelihood to occur in a business environment. Although, the initiatives mentioned above target a number of requirements for the integration of business entities in the digital world; an instrument does not exist to manage and exchange reputation information in the supply chain, which is very critical to form transitory collaborations among partners. One of the targets of iSURF Project is to provide an infrastructure for establishing transitory and dynamic collaboration agreements between trading partners in case of exceptions.

The work presented in this thesis, constitutes "***reputation management***" part of the Transitory Collaboration infrastructure provided by iSURF. On this purpose, a

reputation management model and its exchange mechanism have been realized aiming to address the requirements of an electronic business environment and standards.

In the scope of the thesis work, first we have devised a trust and reputation model in order to reflect the service quality of providers based on the votes of the customers. Up to now there have been a number studies in the literature, aiming to manage reputation information both for centralized and distributed communities. These studies have been examined and reviewed in context of the iSURF Project, which envisions a centralized reputation management authority.

Previous efforts provide a useful basis to realize a reliable trust and reputation mechanism; however there exist issues critical to the reputation management that have not been sufficiently addressed. One of these issues is voting quality of the voters: In the previous efforts, voters are treated as completely honest and ideally acting business partners, which may lead to exploitation of trust values to boost or diminish reputation of a provider unfairly. Another, issue that has not been properly addressed, is the combination of multiple reputation criteria to provide an aggregated trust value. Trust is a multi-dimensional concept, which means competency of partners can be different for different aspects. For example, a provider selling good quality products may have a very bad reputation at on time delivery.

In the thesis work, we developed a model that includes a reliable scoring mechanism and; adaptability to the changes in the environment and providers. In addition to these, we have strived to address shortcomings of the previous studies. In order to tackle non-ideally behaving voters problem, we developed a voting quality mechanism that diminishes the weights of dishonest voters in reputation computation. For the handling of multiple trust criteria, we applied an Artificial Neural Network based solution, combining previous votes of a partner requesting reputation information in order to define the priorities of the partners in reputation search. Thus a reputation mechanism, personalized for the voters has been achieved.

Experimentation results show that, the reputation model presented in the thesis provides a robust and reliable reputation mechanism, and also properly addresses the deficiencies of the previous work.

The developed reputation model could not be effective in reality unless it is used in a business environment. On this purpose, we have also provided an implementation of the reputation model within the building blocks of iSURF Platform. In the implementation, the reputation model has been realized as an application and its functionalities have been exposed to the iSURF community through Web Services. The implementation enables end users to assess the service quality of providers and also search for the partners providing a certain kind of products along with the reputation of the providers. The implementation is integrated with iSURF Global Data Synchronization Service Utility and Negotiation Platform. Interoperability of the implementation with other components of iSURF is based on GS1 XML [7] Documents. GS1 XML Documents do not include any section holding consumer votes and provider reputation. On this purpose, we have extended the relevant GS1 XML documents, in order to cover votes and reputations.

In the next chapters of the thesis, the efforts performed in scope of the study will be expanded. The outline is as follows:

In Chapter 2, information on the enabling initiatives and technologies referred in scope of the thesis will be presented. In this chapter, first we will give details on GS1 [3] and its related standards for identification, message level interoperability and data synchronization. Thereafter the chapter continues with brief information on iSURF project, which defines the scope and requirements of the thesis work. In the last part of the chapter, we skim through the enabling technologies. Chapter 3 elaborates on the previous studies that focus on reputation management. A number of approaches are presented and reviewed, addressing the shortcomings that have to be addressed in this study. Chapter 4 presents the trust and reputation model that has been developed. This chapter covers the requirements of the targeted reputation model; the devised reputation mechanism and experimentation results. In Chapter 5, the implementation of the trust model within iSURF Architecture is detailed. In this

chapter, the data model, software architecture, integration through Web services and user interaction are discussed. Chapter 6 summarizes the effort presented in the thesis and addresses the future work, for which this study can be utilized as a basis.

CHAPTER 2

ENABLING INITIATIVES, STANDARDS AND TECHNOLOGIES

In this chapter, enabling initiatives, standards and technologies are discussed in order to give background information on the building blocks of the thesis work. In the first part of the chapter, information on GS1 is provided. GS1 [3] forms the fundamental of business standards and initiatives in order to establish communication and provide interoperability among business partners in the supply chain. In this thesis work, as well as in the iSURF project, the mediums standardized and initiated by GS1 are extensively utilized and extended when necessary. Section 2.1 aims to give information on these mediums. The chapter continues with description of iSURF Project, in scope of which, the thesis work is conducted. In Section 2.3, brief information is provided on the technologies that are used in the implementation.

2.1 GS1 and Related Initiatives

Standards are agreements that define the rules or guidelines that every stakeholder has to apply and obey. In order to standardize identifications, processes and messages in a supply chain, GS1 (Global Standard One) [3] provides the blueprints that stakeholders have to be conformant with. The building blocks of GS1 are intensively used in scope of iSURF Project as well as in the thesis.

This section aims to provide details on the GS1 System. Section 2.1.1 gives information on GS1 and its standards. The identification keys defined by GS1, which are used for partner and product identification in the thesis, are detailed in

Section 2.1.2. The GS1 XML [7] messages elaborated in 2.1.3, form the standard message format used in the thesis. Section 2.1.4, provides information on the Global Data Synchronisation Network (GDSN) [8] the functionality of which is implemented in iSURF and extended in scope of this thesis.

2.1.1 GS1 and GS1 Standards

GS1 is a global organization aiming to design and implement the global standards for use in the supply chains. GS1 envisions that, standards are the foundation that enables companies to exchange information understandably and clearly, in a globalised economy. By means of standards, costs are decreased for every stakeholder in the supply chain.

GS1 is a global, neutral and ***not-for-profit*** initiative that joins the stakeholders in the supply chain. Companies form all kind of stakeholders (manufacturers, distributors, retailers, hospitals, transporters, customs organizations, software developers and regulatory authorities) are included in the GS1. Under GS1, these stakeholders collaborate to agree on the standards in order to make the supply chains faster, more efficient, simpler and price effective.

GS1 was formed when Uniform Code Council (UCC) [9] and the Electronic Commerce Council of Canada (ECCC) [11] joined EAN International, which was the standards body of Europe. Initially created by suppliers and retailers, aiming to ameliorate distribution of the food and consumer goods to supermarkets, now GS1 standards are currently used by millions of companies in many sectors. In Turkey, GS1 is represented by The Union of Chambers and Commodity Exchanges of Turkey (TOBB) [12] .

The standards and solutions provided by GS1 can be listed as follows:

- **GS1 BarCodes and GS1 Identification Keys:** In order to automatically identify products, locations, assets and logistic units; the standards are provided by GS1 BarCodes. GS1 BarCodes are the most well-known and universally recognizable part of the GS1 system of standards. The

identification in GS1 BarCodes are based on the identification keys of GS1. GS1 Identification keys are detailed in Section 2.1.2.

- **GS1 eCom:** In order to automate transmission of electronic business messages, global standards are determined in scope of GS1 eCom. eCom is based on two components, GS1 EANCOM [14] and GS1 XML [7]. In the thesis, GS1 XML messages are intensively utilized. Detailed information on GS1 XML is provided in 2.1.3

- **GS1 GDSN:** The Global Data Synchronisation Network (GDSN) [8] is an automated, standards-based and global environment to enable synchronous and secure data flow, allowing all partners to maintain consistent data continuously. In iSURF, Global Data Synchronization System (GDSS) aims to provide the functionality provided by GDSN, based on the same standards that GDSN follow. GDSN is a main building block, referred in the implementation of iSURF. Section 2.1.4 elaborates on GDSN.

- **GS1 EPCglobal:** EPCglobal is a new standard that combines RFID, Electronic Product Code (EPC) and existing network infrastructure. EPCglobal aims to leverage identification and tracking of trade items through the supply chain globally.

- **GS1 Traceability:** GS1 Traceability is the GS1 solution to track and trace items in a supply chain.

- **GS1 MobileCom:** GS1 MobileCom, forms the standards for usage of mobile devices in the supply chain.

- **GS1 Upstream Integration:** GS1 Upstream Integration is the solution of GS1, aiming to address the challenges in the supply chain integration between manufacturers and their suppliers.

2.1.2 GS1 Identification Keys

In order to support identification of items, services, locations, logistic units and returnable containers, seven identification keys are determined by GS1 [3] . These keys are:

- GTIN - Global Trade Item Number
- GLN - Global Location Number
- SSCC - Serial Shipping Container Code
- GRAI - Global Returnable Asset Identifier
- GIAI -Global Individual Asset Identifier
- GSRN -Global Service Relation Number
- GDTI - Global Document Type Identifier

GTIN and GLN are the identification keys that have been utilized in scope of the thesis. It is useful to provide further information on these identifiers.

2.1.2.1 Global Trade Item Number (GTIN) and Global Product Classification (GPC)

GTIN identifies any item that can be priced, ordered or invoiced at any part of the supply chain [14]. It is used to retrieve any pre-defined information of a trade item. GTIN is a numeric string that can be 8, 12, 13 or 14 digits long. GTINs are widely encoded into bar codes (the classification of the bar code is determined by the application domain) to be utilized in product information lookups.

Global Product Classification (GPC) [15] is a system that provides a common language to group products in the same way. This common understanding empowers data integrity and accuracy in GDSN. In GPC, products are hierarchically classified into units named: Segment, Family, Class and Brick. Brick is the smallest unit that defines the set of similar products. Every GTIN is maps to exactly one GPC brick,

therefore using a brick GPC in Global Data Synchronisation provides an accurate identification of a product group.

In the thesis, the providers of the products are searched through their GPC codes, in addition to this, the implementation is able to handle GTIN numbers as the product identifier.

2.1.2.2 Global Location Number (GLN)

GLN [16] is the GS1 Identification Key to identify locations. Physical locations and legal entities can be identified through GLN. It is a prerequisite to exchange GS1 eCom Messages [17] and access GDSN. GLN is a 13-digit number, which includes Company Prefix, Location Reference and singe Check Digit. In the thesis, Global Location Number is used to identify the trading entities, which are providers and customers.

2.1.3 GS1 XML

GS1 XML [7] aims to standardize electronic business messages, to leverage speed, accuracy and efficiency of communication among business partners. It forms the messaging standard of GS1 together with GS1 EANCOM [13]. Compared to the GS1 EANCOM, the traditional, EDI [18] based GS1 messaging standard; GS1 XML targets different user groups, thus it is not an replacement for the existing EANCOM messages.

GS1 XML is not a standalone standard that can be thought independent from other GS1 initiatives. For instance, GS1 Identification Keys (GTIN,GLN, SSCC etc.) constitute the identification mechanism of GS1 XML. In addition to this, utilization of GS1 XML in GDSN and EPCglobal, brings the necessity to address the needs of these initiatives.

Development of GS1 XML is based on Global Data Dictionary (GDD) [19], which is the repository of data components and business terms. The data components are created according to the UN/CEFACT [20] Core Components Technical Specification (CCTS) [21]. GS1 XML is developed to satisfy the requirements of a

wide range of sectors; however it is possible to extend existing GS1 XML Messages to meet specific requirements through a well defined methodology. GDSN is an active user of this methodology to extend GS1 XML.

The standard is not bound to any transport medium, thus it is possible to exchange GS1 XML messages on AS1 [22], AS2 [23] protocol or through Web services. Since GS1 XML is an open standard; all the XML schemas can be freely downloaded from the GS1 Website [3].

In the thesis work, the messages exchanged with the Transitory Collaboration Service Utility (TCSU) are GS1 XML Messages. As mentioned above, GS1 XML Messages are extensible, in order to be able to meet specific needs. Since trust and reputation related information is not covered by default, certain GS1 messages have been extended to include customer votes and reputation. The utilization and extension of GS1 XML in the thesis are detailed in Section 5.2.4.1 .

2.1.4 GDSN

Global Data Synchronization Network (GDSN) [8] is an internet based network, enabling companies to exchange and synchronize supply chain data using a standard Global Data Classification (GPC). GDSN is built around GS1 Global Registry and GDSN certified data pools, which are combined to provide secure and continuous synchronization of accurate data. GDSN aims smoother and quicker business processes; accuracy in product orders,; less duplication of systems, data and process and elimination of unnecessary costs. GDSN is based on a publish-subscribe mechanism. As mentioned above, GDSN interconnects trade partners around GS1 Global Registry and certified data pools.

GS1 Global Registry functions as the *"information directory*" of the system, which stores and provides the details of the subscriptions, guaranteeing uniqueness of the registered items and the parties. It also ensures the interoperability of the data pools. GS1 Certified Data Pools are the electronic catalogues providing standardized item data. Master data related with the items are stored and served by data pools. Data Pools can be classified into two categories according to their functionality: Source

and Recipient Data Pools. However in most of the current implementations, the Source and Recipient Data Pool functionalities are provided in a single Data Pool. Before joining GDSN, a data pool has to succeed in Data Pool Interoperability Tests, which are officially conducted by Drummond Group [24].

The flow and interactions among the parties in the GDSN are depicted in Figure 2.1 . (The figure is obtained from GS1 Web Site [3]. The flow of a trade item registration and subscription can be viewed in the figure.

First, the Supplier (seller) loads product data to its Source Data Pool (1). Source Data Pool, stores the bulk part of the registered information itself and registers related part in the GS1 Global Registry (2). When a Retailer (buyer), wants to obtain information on a trade item, it request subscription from its Recipient Data Pool (3). Recipient Data Pool, registers the subscription request in the Global Registry (3). Since, the Global Registry is aware of about registrations and subscriptions, it transmits the subscription request from Recipient Data Pool the Source Data Pool as the result, Retailer is registered to the product information provided by the Supplier. Thereafter, data on the product is published to the Retailer through Source Data Pool and Recipient Data Pool (4), and confirmation on this information distribution is returned back to the Supplier(5). The interactions described are realized through GS1 XML Messages which are detailed in Section 2.1.3.

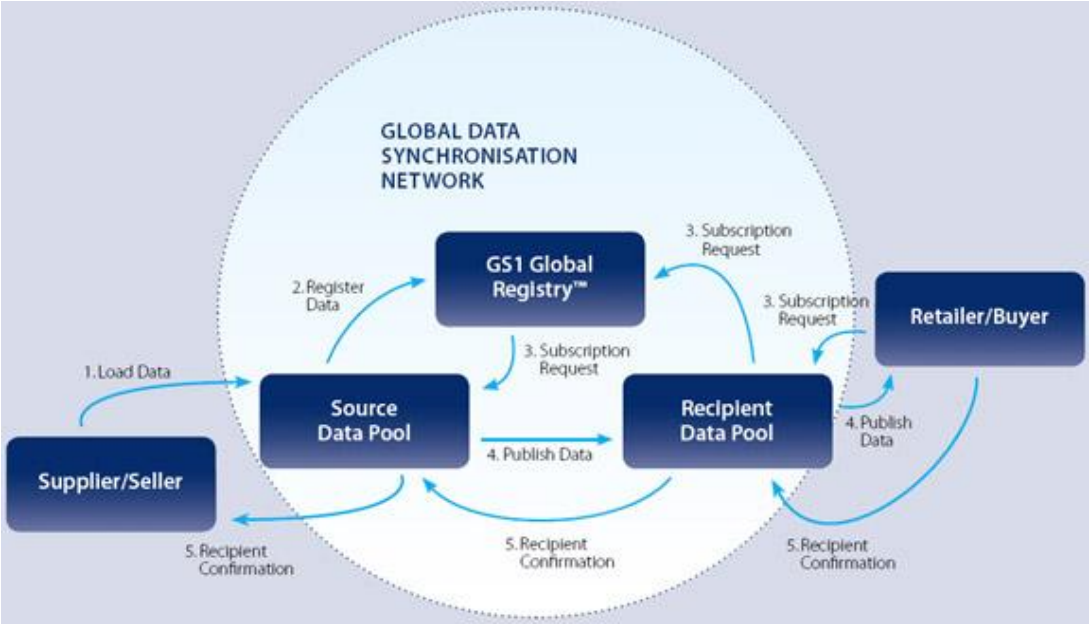

Figure 2.1 - Global Data Synchronisation Network

A trading partner has to register to a Data Pool and pay in order to join GDSN. Since iSURF project aims to leverage data synchronization for SMEs, which may be not able to pay fees for this service, it implements a Global Data Synchronization and Transitory Collaboration Service Utility (GDSSU), which basically functions similar to GDSN. In addition to this, GDSN does not provide any information about reliability and reputation of the trading partners. The work accomplished in the thesis, complements GDSN, providing a trust and reputation mechanism on top of Data Synchronization.

2.2 iSURF Project

iSURF (acronym for: An Interoperability Service Utility for Collaborative Supply Chain Planning across Multiple Domains Supported by RFID Devices)[1] is a research project supported by European Commission Information Communication Technologies (ICT) Seventh Framework Programme (FP7). iSURF aims provide a knowledge-oriented inter-enterprise collaboration environment to European SMEs to share information on the supply chain visibility, individual sales and order forecast

of companies, current status of the products in the manufacturing and distribution process, and the exceptional events that may affect the forecasts in a secure and controlled way. iSURF is a 30-months project started in February 2008. Seven partners from 5 different European Countries are included in the project. The project is coordinated by Middle East Technical University, Software Research and Development Center (METU-SRDC) [25]

In scope of the project, planned work to be accomplished can be summarized as follows:

- Creating "Virtual Organizations" establishing semantic infrastructure for interoperability of enterprise applications across multiple domains, in order to achieve planning data exchange. OASIS UBL [4] will be utilized as the lingua-franca across the domains.

- The Smart Product Infrastructure, a pervasive, dynamic, intelligent and ubiquitous network and service infrastructure to trace RFID-based smart tags within the supply chain in real time.

- A Service Oriented Supply Chain Collaboration environment to coordinate collaborative planning process based on CPFR [6]guidelines.

- A Global Data Synchronization and Transitory Collaboration Service Utility (GDSSU) which ensures the synchronization and harmonization of the master data used in supply chain transactions. On this purpose, A Global Data Synchronization Service(GDSS) to provide Data Synchronization for SMEs, similar to GDSN, has been implemented. For the exceptional cases, Transitory Collaboration Service Utility(TCSU) is implemented in order to replace missing partners with suitable ones.

The software architecture of the iSURF Platform can be viewed in Figure 2.2:

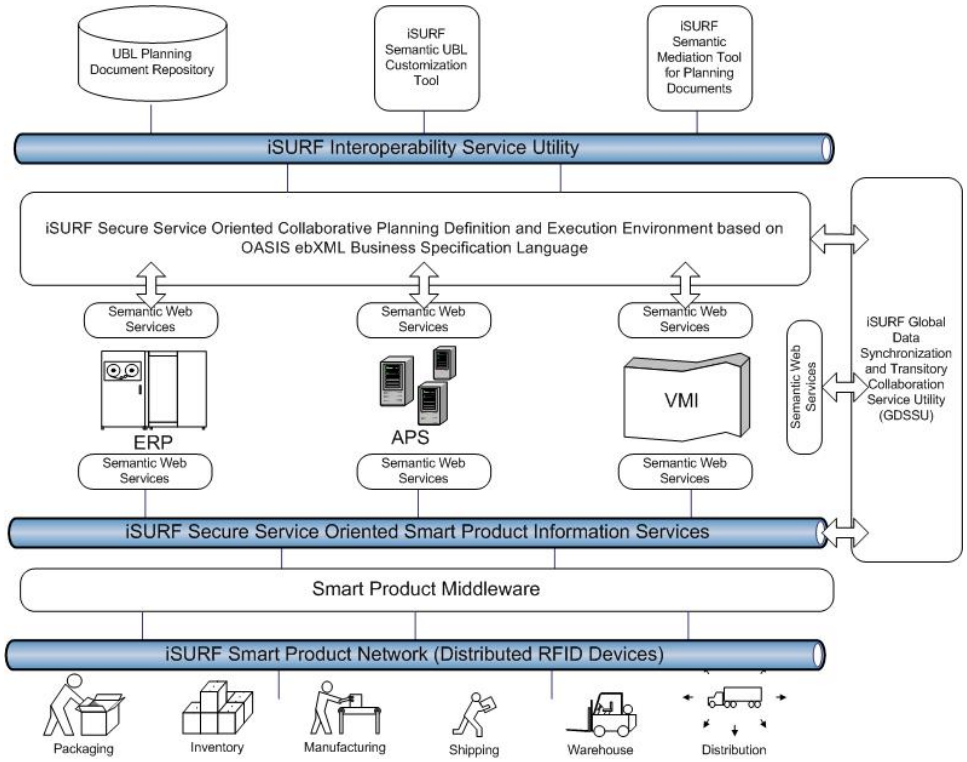

Figure 2.2- iSURF Software Architecture

The thesis work is conducted as the part iSURF GDSSU, aiming to provide a reliable and robust reputation management mechanism for partner search to establish transitory collaborations in exceptional cases.

2.3 Related Technologies

2.3.1 Web Services and JAX-WS

According to W3C definition [26] : *"A Web service is a software system designed to support interoperable machine-to-machine interaction over a network."* Web Services can be seen as application programmer interfaces (API) that can be accessed over a network. While this W3C definition refers to a broad range of applications, the term *Web Service* is used for clients and servers communicating over HTTP protocol through SOAP [27] messages.

SOAP [27] is an XML based lightweight protocol aiming to enable information exchange in decentralized and distributed architectures. A SOAP message consists of three parts which are: an envelope defining the content and receivers; encoding rules defining the serialization mechanism and RPC representation defining the convention for remote procedure calls. SOAP relies on other Application Layer protocols such as HTTP.

Web Services are built on a protocol stack which enables their definition, discovery, implementation and interaction. This stack consists of four layers which are named as: Transport Protocol, Messaging Protocol, Description Protocol and Discovery Protocol.

Transport Protocol is the application layer protocol responsible for the transmission of the messages over network. HTTP is the de facto protocol that is used in Web Services; however other protocols such as SMTP or FTP are also available.

Messaging Protocol defines the encoding of messages into a common XML format that can be understood by both the server and the client. SOAP and XML-RPC are widely used Messaging Protocols in Web Services.

Definition Protocol is the protocol that defines the interface of the Web services WSDL [28] is the typical format to define Web services. WSDL is an XML format that defines Web services (and other network services) as a set of endpoints. In a WSDL, messages and provided operations are defined abstractly and bound to a common message format and concrete network protocol. The public interface of a Web service is defined through this way.

Discovery Protocol is a common registry that is used to locate Web services. In the registry, Web service locations and definitions are stored and published in order to ease discovery of services. Universal Description Discovery and Integration (UDDI) [29] is a registry for businesses to publish their services through internet. However, it is not a widely adopted means of Web services discovery.

In a typical Web service interaction, the provider system (server) implements the service and publishes its interface (WSDL). This can be done either in a ***contract***

first fashion, where the interface is defined first, then the implementation is realized; or ***contract last*** fashion where the service is implemented first and then the interface is defined. The client system implements the codes to invoke the system (either manually or automatically) using the WSDL of the Web service and calls the Web service.

In this study, SOAP Web Services are implemented in order to expose the functionalities provided in Transitory Collaboration Service Utility to the network in a standardized way.

There are a number of available platforms to implement Web services for most of the widely used programming languages. In the study, JAX-WS Reference Implementation [30] is utilized. JAX-WS (The Java API for XML Based Web Services) is a part of Java EE platform for that specifies the Java API for Web services in JSR 224[31] . JAX-WS specifies Java annotations and attributes in order to ease creation of Web Services.

The Reference Implementation of JAX-WS (JAX-WS RI), which is an open source project, is known to be a production quality implementation and appropriate for utilization in applications. For this reason, JAX-WS RI is selected as the Web service implementation medium in the study.

2.3.2 REST

Representational state transfer (REST)[32] is an architectural style for networked systems. The term REST has been coined by Roy Fielding in his dissertation [33]. REST treats Web entities as resources. The resources are represented to the clients and the clients are put in a state. As retrieved representations change, clients change their states. It is analogous to the surfing of a human user to different Web pages. In REST, the services in the Web are viewed as connected network of Web resources.

REST is not a standard thus it does not impose specifications. It is an architectural style instead. It uses a number of well known standards such as URL, HTTP, HTML, XML etc. REST proposes stateless, pull based client-server interaction; uniform interface to access resources (such as HTTP GET or POST); URL naming

of resources and interconnection of resources through URLs and encourages caching of resources.

While XML Web services provide a powerful medium for accessing remote services, it adds extra complexities and dependencies to the system. REST introduces a much more simplified style in order to access remote services therefore they can be seen as lightweight Web Services. In addition to this, REST encourages representation of resources in JSON [34], which is easily consumable format compared to XML for the clients.

In this study, REST is used as an adapter between Transitory Collaboration Web Services and the user interface in order to ease creation and consumption of GS1 documents on the client side. The REST services are invoked through simple HTTP GET and POST requests and return JSON or text data.

2.3.3 JAXB

Java Architecture for XML Binding (JAXB) [35] refers to the Java specification and technology to access XML formats as Java objects. JAXB provides two capabilities which are to marshalling (generating XML from Java objects) and unmarshalling (creating Java objects from XML), the main aim of JAXB is to alleviate the burden of XML parsing in the applications.

In this study, JAXB is utilized in generating Java classes for standard and extended GS1 XML documents; and marshall-unmarshall operations to integrate the system with GDSSU Web services have also been performed through JAXB. In addition to this, JAX-WS Reference Implementation [30] realizes XML generation and consuming through JAXB.

2.3.4 Spring Framework

The Spring Framework [36] is an open source application framework for Java. The framework does not direct any programming model or platform, however it has gained popularity in enterprise Web applications as an alternative to EJB(Enterprise Java Beans) [37] model.

Spring Framework provides a wide range of services, which can be summarized as: Inversion of Control, Aspect-oriented programming: Data access, Transaction management, Model-view-controller: Remote Access framework, Batch processing, Authentication and authorization, Remote Management , Messaging and Testing

In this study, Spring Framework is mainly used for Inversion of Control, Data Access and Testing.

Inversion of Control (IOC) [38] is a principle to decouple the implementations from the interfaces of the objects. In Spring IOC container, the Java object implementations are accessed through interfaces which are defined in XML configuration files and injected into the application when needed. Dependent classes do not have to know about the actual implementations therefore the application is highly decoupled.

Data Access Services of Spring Framework provides abstractions and functionalities to leverage access to databases and interaction with Object-Relational Mapping [39] Tools.

In order to test the core functionalities, unit tests have been coded using the testing services of the framework.

In this study, Spring Framework is utilized in Transitory Collaboration Web Services and REST Services implementations.

2.3.5 JPA and Hibernate

The Java Persistence API (JPA)[40] is the API to map Java entities into relational models. JPA provides a standard way for Object-relational mapping [39] and simplifies methods to store the entities. Through its Java annotations, JPA eliminates the necessity of metadata to persist entities and extends the capabilities of the present object persistence methodologies. JPA is a specification and does not direct any specific implementation. Applications can choose any JPA compliant implementation to realize persistence. The Java Persistence API has been created as part of the work of the JSR 220[42] Expert Group.

Hibernate [41] is an open source persistence implementation for Java the aim of which is to map Java entities into relational models. Starting from the version Hibernate 3, it is a certified implementation of JPA specifications.

In this study, standard JPA annotations are utilized to define the persistent Java entities in the Transitory Collaboration Service Utility. To realize persistence, Hibernate version 3.3.2GA has been selected as the JPA implementation. In the system, Hibernate has been utilized through the data access services of the Spring Framework [36].

2.3.6 JSON

JavaScript Object Notation (JSON)[34] is the lightweight data interchange format which is described as more readable for humans and easier to process for the machines. It is a language independent text format that is used as a data interchange language.

JSON is built on two structures: collection of key value pairs(similar to hash tables) and ordered list of values (similar to arrays), which are universal data formats that are supported by every modern programming language.

In this study, JSON format is used as the return format of the REST Services since consuming complex XML documents can be a tedious process for a platform other than Java (such as Flex). In REST Services the relevant part of the requested data is extracted from GS1 BMS XML documents to JSON so that they are easily processed on the client side.

2.3.7 Adobe Flex

Adobe Flex [43] is a free open source framework for creating and maintaining rich internet applications (RIA). Applications can be developed using only the Flex Software Development Kit (SDK) however a commercial builder is also provided to ease GUI development. Flex applications are built on top of the Flash [44] Platform and run on the browser of the client.

Flex provides a stateful client environment in contrast with the classical browser applications which eliminates the requirement to reload the pages. The client application can be connected to the server through a number of ways. HTTP request is a common way to provide this interconnection. In Flex, graphical user interfaces are defined within MXML [45] format and logic is implemented through ActionScript [46]scripting language.

Adobe Flex has been utilized to create the graphical user interface of the system developed in this study.

2.3.8 Joone

Joone [47] is an open source, component based artificial neural network engine. It provides an extensible Java based Neural Network Engine which can be accessed programmatically in order to create, train and query neural networks. It provides various learning algorithms, propagation mechanisms, integration with data sources, and distributed computing facilities. Its extensibility enables integration with new data sources, applications and programming mediums. A graphical user interface is also provided to work with the neural networks.

In this study, Joone Engine is utilized in combining multiple trust criteria for determining the reputation of a provider, personalized for the requestor.

2.3.9 MySQL

MySQL [48] is an open source Relational Database Management System (RDBMS) that is known to be the most common open source database system in the world. MySQL is widely used as the database component of many popular Web applications such as Google, Flickr, Facebook[49].

In this study, MySQL is used to persist entities in iSURF TCSU. As mentioned above, the persistence is provided through JPA.

2.3.10 Apache Tomcat

Apache Tomcat [50] is the open source servlet container which is an implementation of Java Servlet [51] and JavaServer Pages (JSP) [52] specifications. It provides a pure Java HTTP Web Server Environment for Java Code to run.

In the study Apache Tomcat Version 6.0 is used to host Web and REST services. In addition to this, user interface is deployed and provided within a Tomcat Server.

CHAPTER 3

RELATED WORK

There have been a number of studies to manage reputation of agents in electronic environments, for both centralized and P2P architectures. Since iSURF is built on a centralized system to share master data among organizations; the studies on centralized architectures have been taken into account primarily, however the P2P related efforts have been referred as needed.

The widely known electronic market places, eBay [53], Amazon Auctions [54] and OnSale Exchange [55] provided simple online reputation mechanisms to prevent cheat and fraud. In eBay, users submit vote (-1, 0, +1) for their partners. The reputation of an agent is determined by the received sum of votes for the recent six months. *Amazon* and *OnSale* utilize the arithmetic mean of the received votes to assess reputation of an agent.

3.1 SPORAS and HISTOS

SPORAS [56] proposes a reputation mechanism that relies on collaborative ratings and personalized evaluation of the various ratings assigned to each user. The following concerns have been taken into account in *SPORAS*:

- Users with bad reputations may try to re-enter to the community with new identities.
- Users with bad reputations may perform fake transactions to boost reputation of each other.
- Users with fake identities may vote for an agent to boost its reputation.

- Since the human factor is involved in reputation, the users may change their behavior in time. Therefore, the old votes for a user have to be discarded.

Based on these considerations, SPORAS has been implemented based on the following principles:

1. New users start with a minimum reputation; they build their reputation through the interactions in time.

2. The reputation of a user can never fall below the reputation of a new user.

3. After each vote, the reputation value of the votee is updated based on the rating received from the user.

4. A user may vote one user only once. If there are more than one ratings from one user to another, the most recent one is taken into account. In this way, the effect of fake transactions to boost one's reputation is avoided.

5. Users with high reputation are affected less compared to the ones with low reputations. Thus malicious attacks on the highly reputed users are less effective. However, SPORAS does not protect reputed users indefinitely.

6. Reputation of a user cannot increase infinitely.

In SPORAS, the reputation of a user is updated according to the following formula (3.1):

$$R_{t+1} = \frac{1}{\theta} \sum_{1}^{t} \Phi(R_i) \; R_{i+1}^{other} \quad (W_{i+1} - E(W_{i+1}))$$

$$\Phi(R) = 1 - \frac{1}{1 + e^{\frac{-(R-D)}{\sigma}}} \quad and \quad E(W_{t+1}) = R_t / D \tag{3.1}$$

In this formula:

R_t : reputation of user at time t.

R^{other}: reputation of the user giving the rating.

Φ: the damping function to minimize the effect on high reputations.

σ: the acceleration factor of the damping function Φ

θ: a constant integer greater than 1. It determines the effect of each vote, which can be viewed as a kind of memory in the system.

D: range of votes

W: rating given to the user

According to the authors, results show that SPORAS performs well in exhibiting the reputation of users at different quality level. In addition to this; it is robust to the changes in user behavior in time.

However SPORAS takes the reputation of the voter into account, which is not a realistic approach since a reputed user might not be a good or honest voter. In addition to this, multiple criteria has not been taken into account in SPORAS, although its essentiality is pointed by the authors.

HISTOS [56] is an extension to the SPORAS aiming to personalize the reputation model. In this approach, reputation of a user is personalized for the querying user, utilizing the ratings given by witnesses. The witness ratings are weighted according to the reputation of the witnesses which is calculated recursively. HISTOS provides a more personalized approach however, it suffers from the same deficiencies that SPORAS have.

3.2 REGRET

REGRET [57] is the acronym for "A REputation Model for GRegarious SocieTies". It is a modular trust and reputation system for e-commerce environments for which relations among individuals play significant role.

REGRET defines the reputation according to the following formula (3.2):

$$R^t(IDB_p^a) = \sum_{\iota_i \in IDB_p^a} \rho(t, t_i) \cdot W_i \tag{3.2}$$

In the formula:

R(IDB): reputation of a user *a* with respect to the voters in set *p*

W_i: the rating given to user a in the vote i.

p: the time dependent weight function for vote.

As seen in the formula, the weights of votes changes in time. Authors propose a simple function of time t_i/t to diminish the effect of old votes. In addition to this, after a constant duration, the old votes are totally discarded. This formula is usually written as the following shorthand form (3.3):

$$R_{a \to b}(subject) \tag{3.3}$$

In REGRET, the reputation information is retrieved from different sources of data which are:

- **Individual Dimension:** The reputation is calculated based on the direct interactions of agents. It can be viewed as the personal experience of the user.

- **Social Dimension:** In addition to the personal experience, the social group that user belongs to plays a significant role in reputation evaluation. In social dimension of REGRET, three kinds of interactions are taken into account**:**

    - **Personal Experience:** Personal Experience refers to the individual experience of a voter with a single votee and the group of this votee. The first one is covered with the individual dimension whereas the latter is defined as follows(3.4):

$$R_{a \to \mathcal{B}}(subject) = \sum_{b_i \in \mathcal{B}} \omega^{ab_i} \cdot R_{a \to b_i}(subject) \tag{3.4}$$

    - **Group Experience:** Group experience defines two kinds of experiences. First one is the reputation of the votee agent for the voter's group. It is defined in (3.5)

$$R_{\mathcal{A}\rightarrow b}(subject) = \sum_{a_i \in \mathcal{A}} \omega^{a_i b} \cdot R_{a_i \rightarrow b}(subject) \quad (3.5)$$

The second one is the experience of the voter's group with the votee's group. The definition can be viewed in (3.6):

$$R_{\mathcal{A}\rightarrow \mathcal{B}}(subject) = \sum_{a_i \in \mathcal{A}} \omega^{a_i \mathcal{B}} \cdot R_{a_i \rightarrow \mathcal{B}}(subject) \quad (3.6)$$

- **Ontological Dimension:** In REGRET, authors state that reputation is a multi-faceted (multi criteria) concept which means that reputation has to be calculated for different criteria. Authors classify the reputation criteria through ontologies and combine the aggregated reputation as a function of the single calculated reputations. Again a linear combination is proposed.

REGRET has achieved to provide a robust reputation management mechanism to handle social dimension of trust with small number of interactions. However, the behavioral aspects such as dishonest voters or changing service quality have not been taken into account deeply. In addition to this, mechanisms to combine the multiple criteria have not been discussed in detail.

3.3 DIRECT

DIRECT [58] is the reputation model proposed in scope of the FP7 project ECOLEAD [59]which aims to induce creation of Virtual Organizations (VO). Authors have emphasized distributed reputation management, dynamism to quickly

form trust reputations, adaptability to the changes in the environment and predictability as the requirements of a reputation system.

DIRECT reputation model is based on the concept of *Direct Reputation* (DR), which is the weighted average of assessments made on one agent's competence. The authors view *Direct Reputation* as a context dependent issue which means that trust cannot be defined as an isolated, generic concept.

In DIRECT reputation model, reputation values are numbers in the range [0, 1] different from the SPORAS trust model. In addition to this, a recent time window (W) is defined so that, only the interactions in W (most recent ones) are taken into account and the older ones are discarded.

Reputation is the composition of the single assessments named as *impression.* Impression is a single dimensional mathematical value which is defined in (3.7):

$$imp_{ij}^{d} \in [0,1], \tag{3.7}$$

In the formula, *i* and *j* denote the voter and votee whereas *d* is the context for which the impression is acquired. The values are again in the [0,1] range as in the calculated reputation.

The context defined in DIRECT, can be either an objective criterion such as delay in the delivery or a subjective idea such as consumer's empirical estimation for the quality of service. The values can be mapped to impressions according to predefined thresholds.

DIRECT reputation is computed and updated incrementally according to the formula in (3.8):

$$DR_t = DR_{t-1} + \alpha \cdot [imp_t - DR_{t-1}]. \quad (3.8)$$

In the formula, DR_t is the reputation value at time t where imp_t defines the latest impression. α is the shorthand term for the function $\alpha(DR_{t-1}, imp_t)$ which indicates the memory of the system. As α increases, the memory of the system is decreased. The initial value of α where t is 0 is 0.5. α is a recursive function defined in (3.9):

$$\alpha(DR_t, imp) = \frac{\alpha(DR_{t-1}, imp) + \beta(DR_{t-1}, imp)}{2}. \quad (3.9)$$

In the formula, β is the similarity function defined with (3.10) and (3.11):

$$\beta(DR_{t-1}, imp_t) = 1 - e^{-10 \cdot ABS(E - imp)} \quad (3.10)$$

$$E = \frac{DR_{t-1} + imp_{t-1}}{2}. \quad (3.11)$$

Here E is the average of reputation and most recent impression which is utilized in computation of the similarity between expected reputation (DR_t) and the impression (imp_t).

DIRECT is compared with SPORAS and REGRET in terms of accuracy and adaptability to behavioral changes of the service providers. According to the authors, results show that DIRECT is as robust as REGRET at computing the reputation

accurately with small number of interactions. Also it is as adept as SPORAS to adapting to the changing service qualities of the providers.

However, handling of multiple dimensions of reputation (context in DIRECT's terminology) and the behaviors of the voters are left open and have not been addressed in the study. These aspects need further attention and improvement in order to achieve a functional and realistic reputation system.

3.4 Other Work

In this section, other relevant previous work on trust and reputation are briefly mentioned. Most of these studies are applicable for only distributed architectures thus are not directly related with the requirements of iSURF Project.

In Yu's study [60], trust management has been considered as a social issue and a complementary means to security mechanisms. The study mainly aimed decentralized environments in which a central authority is not always readily available. Yu emphasizes the impressions of the witnesses who have interacted with the agent the reputation of which is being calculated. In addition to this, propagation, gossip and other heuristics are taken into account.

In TrustGuard [61], mechanisms to handle fake votes, fake transactions, vote manipulations by malicious node groups and fluctuating behavior changes in P2P environments are elaborated. In order to weigh witness reputations, Probability Similarity Measure (PSM) is utilized. PSM indicates the similarity of a witness to the agent which makes trust assessment. PSM is a noteworthy approach which can be considered in calculation of weight of the votes.

Beta reputation system[62], uses the beta distribution [63] which is asserted as a simple yet efficient method to calculate reputation.

Liang and Shi [64] state that using complex algorithms such as PSM are not superior to the simplistic ones such as Beta, therefore for the environments with limited computational resources (such as P2P), simpler approaches should be preferred.

CHAPTER 4

TRUST AND REPUTATION MODEL PROPOSED IN THIS THESIS

This section briefly explains the importance of Trust and Reputation and elaborates our approach which aims to tackle a number of shortcomings in the similar previous work. Thereafter, the simulation results for the trust model we have developed are revealed and discussed. The last part of the section is dedicated to the description of multiple trust criteria and their fusion in iSURF context.

4.1 Overview

By definition, "trust" refers to the assured reliance on the character, ability, strength, or truth of someone or something [65]. It is not only an estimation of intention but also a measure of possible competence and deception for establishing relations [66]. According to Gambetta [67], "Trust is the subjective probability by which an individual A, expects that another individual B, performs a given action on which A's welfare depends."

Trust has been a research area for different disciplines: sociology, economics, psychology and politics. From the aspect of social sciences, trust is based on the previous experiences and interactions among individuals, and it is heavily affected by random and non-deterministic factors. It is slowly built, but can be easily destroyed. From the economic point of view, it is a measurement of taking a risky choice in order to maximize gain.

The first studies on trust in Computer Science have focused on: Security and Privacy. A trusted environment generally refers to an environment in which access to sensitive data is controlled. A number of protocols and mechanisms have been realised to assure Trusted Environments.

Computational Mechanisms for trust and reputation in virtual societies have been a recent area of research in order to foster reliability and performance in electronic communities [68]. Development in "Multi Agent Systems (MAS)" and the evolution from the paradigm of isolated machines to the paradigm of networks have boosted the interest in the area.

In virtual communities, such as e-commerce and social networks, "reputation" is an efficient and ubiquitous key factor in establishing relations that has to be maintained and managed. Reputation Management includes recording an agent's past actions, interactions and the opinions of the other agents on these interactions. These records can be either publicized to the community, or can only be used as an indicator of reputation of an agent to help others in deciding whether to trust or not to that agent.

In some of the electronic communities, the architecture involves a centralized repository, which allows reputation management within a central body; whereas in P2P communities there is no central repository to maintain reputation and the agents are dynamic which are expected to enter or leave the community any time. In these architectures, each agent maintains the necessary trust information itself. In addition to this, in P2P architectures, the cost of computation has to be more deeply taken into account in order to avoid performance problems.

The reputation management in this thesis is built on top of a centralized architecture, in which iSURF TCSU is responsible for all of the flow and maintenance of trust and reputation related data.

4.2 Trust Model

In this section, the trust model proposed in this study has been detailed. The model aims to tackle a number of open issues and shortcomings exhibited in the previous

studies. In the first section, the motivation and the considerations behind the implemented model have been revealed. Later on details of our approach have been provided. In the following subsections, the outcomes of the experimentations have been presented and the results are discussed. The results obtained from the show that, a successful model have been achieved tackling the issues that we have focused.

4.2.1 Considerations

The trust model is designed to be a function within the deployment of iSURF together with iSURF GDSSU, and iSURF TCSU. Therefore, the requirements of the application domain which may not be covered in the preceding studies had to be considered and addressed in the model. In order to be operational and robust, following aspects have been taken into account.

- **Centralized Architecture:** In iSURF, there exist central components which are designed to be always up and running. Taking this into account, our trust model is based on a centralized architecture instead of a distributed one. Therefore all the assessments and storage of reputation values are conducted on a centralized body (iSURF TCSU in the actual implementation). In addition to this, the target environment is not highly dynamic; therefore fake transactions and voters with fake ids are not likely to occur. Hence, they have not been considered in the model.

- **Accurate results with small number of interactions:** A reputation management system should provide accurate results with a small number of votes in order not to cause misjudgments. In SPORAS, it takes a number of interactions to reflect the real service quality. On the other hand, REGRET and DIRECT use the average of the votes, thus they tend to reflect realistic reputation scores with fewer interactions. In our model, the approach of REGRET and DIRECT has been followed and the votes are directly included in the computation.

- **Adaptability to the changes in service quality:** In a realistic environment, it is highly probable that the service quality of a provider will change in time. For such situations, the reputation system has to quickly adapt the new quality and evaluate the trust accordingly. In SPORAS and DIRECT, this requirement is addressed, however in REGRET, it is not. In the study, we addressed this requirement, discounting the weight of the votes by time.

- **Taking non-optimal behaviors of the voters into account:** In a realistic environment, it is hard to expect that all the voters will vote the other agents in a completely honest fashion. Either intentionally or not, the votes will deviate from the actual service quality. Both mean and over generous voters may exist in the system. In our model, we have assumed that the majority of the voters tend to assess honestly, whereas some of the voters are inclined to assess far from the actual quality. Based on this assumption, we have strived to make reputation values closest to the votes of the majority. In the previous studies, the non-optimal behaviors of the voters have not been addressed in detail. In SPORAS, the reputation of the voter is taken into account however this is not a realistic approach since a reliable service provider may not be a reliable voter or vice versa. In TrustGuard, the PSM algorithm favors the assessments of other agents which exhibit proximity with the agent making the evaluation; however this is a different issue than what we have tackled in our model.

- **Ability to combine multiple trust criteria:** A number of previous studies have mentioned multi-dimensional nature of trust and reputation but, except REGRET these studies do not address mechanisms to combine multiple dimensions to aggregate a single reputation value. REGRET proposes an ontological view to handle multiple criteria however it does not address a methodology to weigh these criteria. In our model, we have combined multiple criteria, weighing them according to the previous votes that the party requesting the reputation score had submitted. Details of the aggregation have been presented in Section 4.3.

4.2.2 Approach

This section elaborates on the details of the reputation system based on the considerations revealed in the previous sections. As mentioned above, handling of multi-dimensional criteria will be discussed in Section 4.3, thus computation of trust in single dimension will be detailed in this section.

4.2.2.1 The Trust Formula:

In the trust model, we adopted a trust formula that resembles the one in REGRET, in order to provide meaningful outcomes in small number of interactions. Thus, the impressions are immediately reflected as reputation values. However, the weighing of the single impressions (votes) is quite different than REGRET's approach. In (4.1) the general formula to calculate reputation can be viewed.

$$R_a = \sum_{v \in V_a} w_v \times s_v \tag{4.1}$$

In this formula, R_a is the reputation of the provider a at a time. V_a is the set of votes that a has received; whereas v is a single element of this set. The terms on the right side of the equation are w_v and s_v. The latter is the score given to the provider which is in the range [0,1]. w_v is the weight of the vote.

In weighing of the votes, two criteria have been taken into account, which are: time weight and vote quality. Weight of a single vote is described in (4.2):

$$w_v = \frac{tw_v \times qw_v}{\sum_{vote \in V_a} tw_{vote} \times qw_{vote}} \tag{4.2}$$

In the formula, the sum of all weights is equal to 1 and all of the votes given to a provider a are taken into account. As mentioned above, there are two factors determining the weight which are time weight (tw) and quality weight (qw) or voting quality. Following sections explain the calculation of these factors.

4.2.2.2 Time Weight

In a realistic e-commerce environment, the providers may change their service quality in time. They may serve with a much high quality compared to initial state, or they may reduce service quality trying to exploit their older reputation. A reputation system has to be robust enough to handle both of those issues.

In order to adapt the new conditions faster, the weight of old votes have to be decreased so that recent impressions can gain higher importance. The time weight of a single vote *v* is computed as in (4.3):

$$tw_v = \begin{cases} 0 & t_{current} - t_v > t_{MAX} \\ 1 - \dfrac{t_{current} - t_v}{t_{MAX}} & otherwise \end{cases} \tag{4.3}$$

In this formula, $t_{current}$ is the current moment, at which the reputation is computed and t_v is the time at which the assessment has been made. t_{MAX} is the size of most recent votes window, which is the storage duration of the votes. t_{MAX} can be viewed as the memory of the system. As t_{MAX} increases, the memory of the system will be longer.

The time weights of the votes diminish linearly until t_{MAX} is reached. After t_{MAX}, they are totally discarded. Thus, old votes will not remain in the system indefinitely.

4.2.2.3 Voting Quality:

As mentioned in the considerations, it cannot be expected that all voters in the system will behave ideally in a realistic environment. Some voters may be too mean in assessing the others and assess lower scores than the actual quality whereas some others may overrate the providers and assess too generous votes compared to the actual quality. As the number of non-ideal voters increase in the system, the trust model will suffer in terms of accuracy.

In order to tackle this problem, we have introduced the parameter, "*voting quality"* to reflect the behavior of the voters in the system. Quality of a voter is determined as the time weighted average of proximity to other voters' assessments. As aforementioned, here it is assumed that the majority of the voters act honestly, therefore the generous and mean voters are regarded inferior and their weights are tried to be diminished.

Quality of a voter x, q_x is determined as in (4.4):

$$q_x = 1 - \sqrt{\sum_{v \in V^x} wt_v \times \left(s_v - recent\_score(provider_v)\right)^2} \qquad (4.4)$$

In this formula, V^x is the set of votes assessed by *x,* and v is a single element in the set. *wt* is the time weight of a single vote which can be obtained with (4.5) (the calculation of *tw* had been explained in the previous section):

$$wt_v = \frac{tw_v}{\sum_{vote \in V_a} tw_{vote}} \qquad (4.5)$$

s_v is, as mentioned above, the score given in the vote. The function *recent_score* is the arithmetical mean of the scores that a provider has received in a recent time window (which is assumed to be smaller than t_{MAX}). The formula resembles the Euclidian distance, aiming to formalize the distance from the majority of the voters.

The quality of the voters (q_x) provide a reliable measure to denote the voting qualities of the voters, however placing it directly in (4.2) does not yield to the expected effect since the quality values of the voters are close to each other. In order to scale the difference of the voter qualities and obtain *qw*, we applied (4.6) to quality values.

$$qw_v = \frac{q_{voter_v} - q_{MIN}}{q_{MAX} - q_{MIN}} \tag{4.6}$$

Here q_{MIN} and q_{MAX} refer to the minimum and maximum voting quality of the voters in the system. Applying the formula, the voting qualities in the system are scaled along the range of [0, 1], which provides a sufficient span to differentiate optimal and mean voters.

4.2.3 Experimentation

In order to test the validity of the model presented above, we have conducted a set of experiments through simulation for a single dimension of reputation. In the simulation, some assumptions have been made in order to conform to requirements of a realistic scenario.

In the simulation, there are four service providers with the qualities 0.25, 0.50, 0.75 and 1.00. These providers serve the voters (customers) with a maximum 25% deviation from their actual qualities (The qualities exhibit a normal distribution around the actual quality). In order to measure the effects of change in the service

quality, the service qualities of each provider are decreased by 50% in some of the experiments.

There are two types of voters in the system, which are named as HONEST and MEAN voters. HONEST voters assess the actual received service quality whereas MEAN ones assess half of the received service quality. Generous voters have not been included in the simulation since their effect is virtually same as MEAN voters'. The total number of voters is 40.

The simulations have been performed as series of iterations. In terms of a realistic scenario, iterations are analogous to calendar months. During the run of simulations, in every iteration, two voters are selected randomly and they evaluate all of the providers. After iteration is complete, reputation of the providers and voting qualities are re-calculated.

In the simulation, t_{MAX} is selected as 18, i.e. votes are discarded after 18 iterations (18 months in other words), *recent_score* of a provider is the average of scores received in last 4 months.

In the simulation four kinds of experiments have been conducted. In the first experiment, an environment with constant service quality and honest voters is set. In this experiment, we tried to prove that our model performs well under optimal conditions. In the second experiment, the service qualities remained constant; but mean voters have been added to the system. This experiment aims to test the effect of dishonest voters. In the third experiment, an environment with honest voters and providers changing their service quality has been set. After 66^{th} iteration, the service qualities of all providers have been halved. In this experiment, ability of our approach to adapt changing service qualities has been explored. The last experimentation is performed changing service quality of the providers and having a number of dishonest voters in the system. In this experiment, we tested the robustness of our trust model in a more realistic and non-optimal environment.

We compared our results with SPORAS and REGRET, re-implementing those algorithms. In our implementations of REGRET and SPORAS, we fixed the

constants and tailored the algorithms considering relevance and consistency with our model.

The simulations had been run with no less than 150 iterations and repeated 30 times.

4.2.3.1 Experiment 1(Constant Service Quality; 40 HONEST voters)

In this experiment, we used providers with constant quality and ran the system with 40 honest voters; our expectation was having similar results with REGRET and to give accurate scores in less number of iterations than SPORAS.

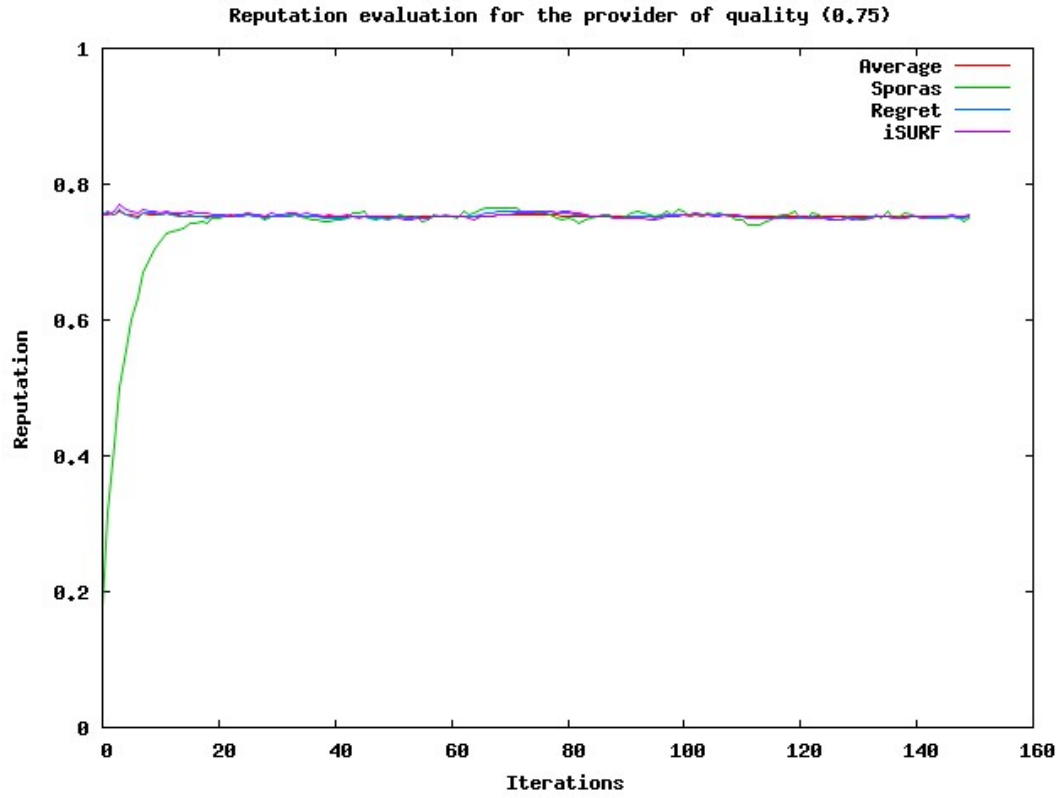

Figure 4.1- Experiment 1 result for provider with quality 0.75

The outcomes of the experiment are consistent with the expectations. As seen in the example, for the provider with quality 0.75, both REGRET and our model give accurate results in small number of interactions. However, SPORAS needs more

than 20 iterations to catch up with other algorithms. The result shows that our approach is valid for stable providers and honest voters.

4.2.3.2 Experiment 2(Constant Service Quality; 24 HONEST, 16 MEAN voters)

In the second experiment, 40% of the voters (16) have been selected as MEAN (i.e. they assess half of the delivered service quality). Our expectation for our model was having closer reputation scores to the actual qualities compared to REGRET and SPORAS. A sample experimentation result is displayed in Figure 4.2.

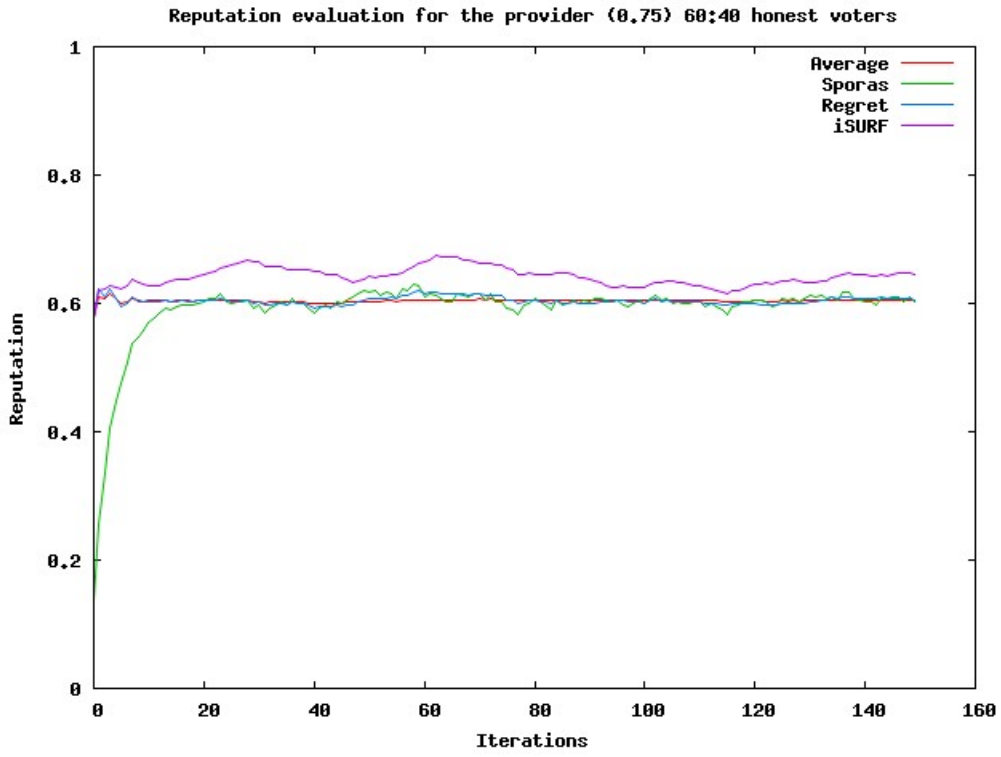

Figure 4.2 - Experiment 2 result for provider with quality 0.75

The results show that, in an environment with 40% mean voters, REGRET and SPORAS tend to give around 0.60 for a provider of quality 0.75. The score given by our model is around 0.65, which is much more close to the actual quality. This

suggests that our model provides more accurate results in an environment of heterogeneous voters.

4.2.3.3 Experiment 4(Changing Service Quality; 40 HONEST voters)

In this third experiment, providers with changing service quality have been used in an environment where all voters are honest. Providers change their service qualities diminishing them by 50% at the 66th iteration. The expected result for our model is to adapt the new service qualities of the providers no later than SPORAS and REGRET. Needless to mention, accuracy in reflecting the qualities of the voters is also compulsory. A sample experimentation result is displayed in Figure 4.3

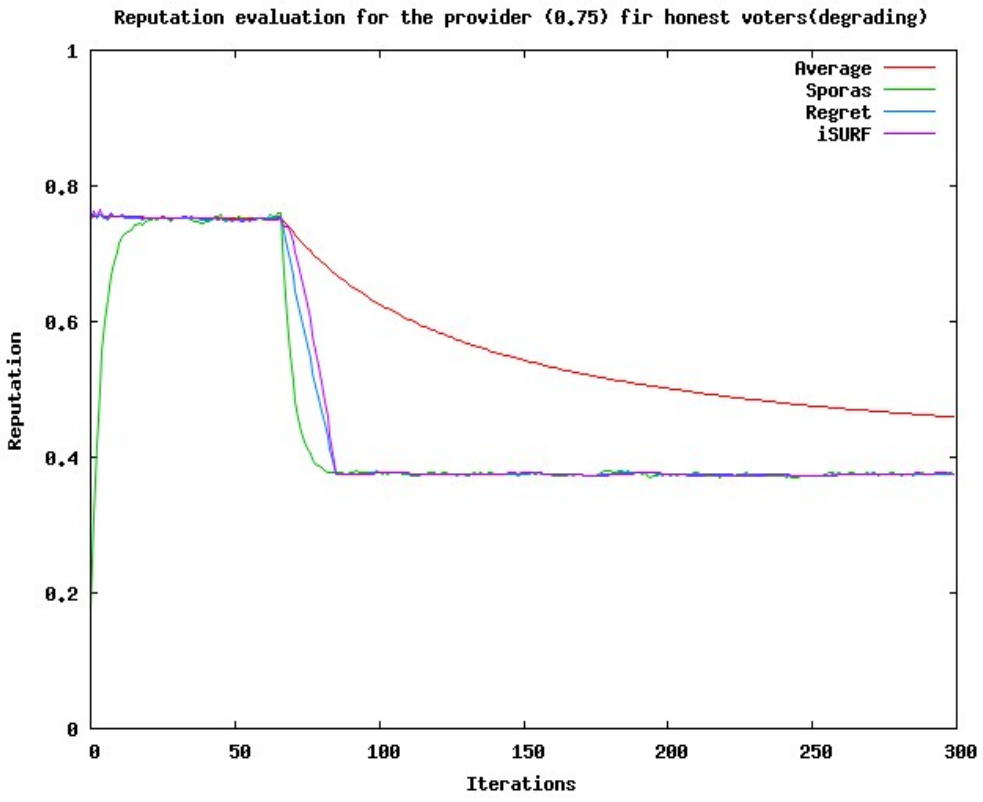

Figure 4.3- Experiment 3 result for provider with quality 0.75

The results show that all of the three algorithms can reflect the qualities accurately whereas SPORAS adapts to the changing service qualities of the providers faster than DIRECT and our model. However the performance of DIRECT and our model are comparable with SPORAS, therefore it can be deduced that the approach that the model proposed in the thesis is valid for the completely honest environments with altering service qualities.

4.2.3.4 Experiment 4(Changing Service Quality; 24 HONEST, 16 MEAN voters)

In this last experiment, the providers diminish their service quality at the 66th iteration by 50%. As in the second experiment, 40% of the voters have been selected as mean voters. The expected result for our model is to give more accurate results compared to SPORAS and REGRET. In addition to this, our model is expected to adapt the new conditions in a comparable speed with the other two algorithms. A sample result can be viewed in Figure 4.4.

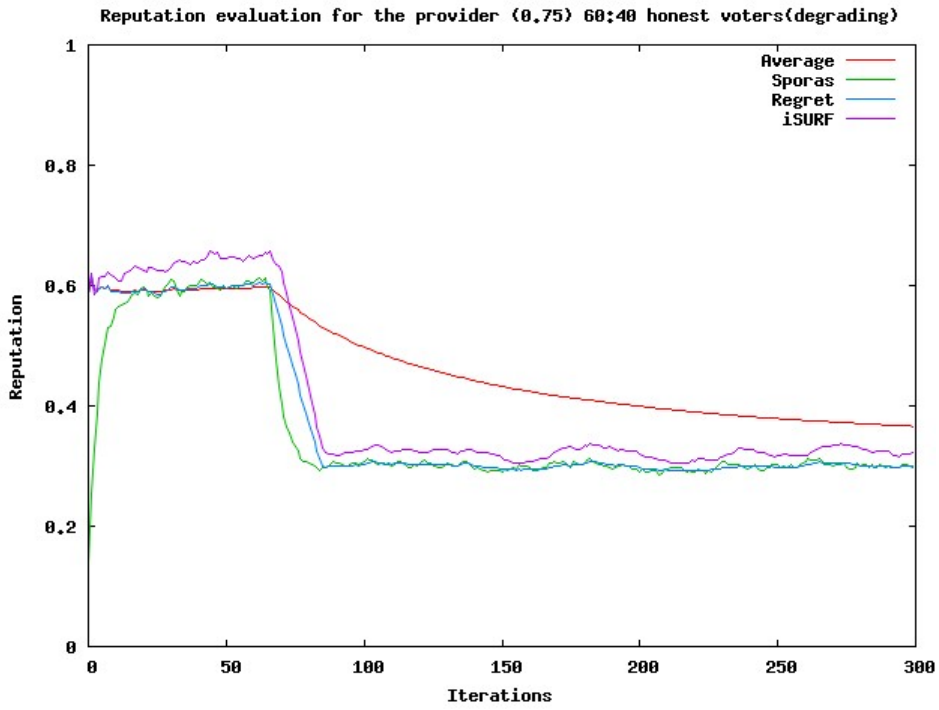

Figure 4.4 - Experiment 4 result for provider with quality 0.75

The results show that our model provides more accurate results than other algorithms both before and after the quality change, similar to the previous example. our estimates around 0.33; SPORAS and REGRET estimate around 0.30 when the actual quality is 0.38. On the other hand, SPORAS adapts faster than our model and REGRET, however the settling down of the new scores of SPORAS takes almost as many iterations that as our model and REGRET need.

The third experimentation proves that, the model proposed provides reliable results in case of changing service qualities and heterogeneous voters.

4.2.4 Results and Discussions

The experimentation results show that, our approach provides an accurate yet fast adapting trust model in order to calculate reputations. The results are not limited only with optimal environments. In the environments with providers changing service qualities and non-ideal voters, our approach gives more accurate results than other relevant approaches. In addition to this, our approach adapts the changing service qualities no slower than the other two algorithms.

In addition to this, our model brings no extra computational cost compared to the other algorithms. The computational efficiency of our approach addresses the concerns stated in Liang and Shi's study [64].

Solutions to adapt quality changes in less number of iterations constitute a significant future work of this study. On this purpose, service quality trends can be detected or ad-hoc approaches can be applied in order to perceive quality changes. The model can also be improved in terms of accuracy to obtain more accurate results. This can be done through revising and enhancing the voter quality computations.

4.3 Multi-Dimensional Trust in iSURF

As mentioned in Section 4.1, trust is regarded as a multi-dimensional concept. This multi-dimensional aspect is valid in also iSURF domain. A price efficient seller may be bad at delivering on time, or may provide low quality products. Previous studies in the field do not address the multi-dimensional criteria in detail even though they mention its importance. In REGRET, an ontological dimension of trust is introduced however any effort to automate the combination of ontological dimension aspects has not been made.

In this study, we propose a personalized approach to combine the multiple trust criteria in order to generate a single reputation value. Our approach aims to detect the priorities of the customers (voters) and generate reputation values that satisfy their preferences.

There are three dimensions of reputation in iSURF, which are ON TIME DELIVERY, PRICE and QUALITY. In order to detect user priorities, we added another vote type which is OVERALL score. Here, two assumptions have been made which are as follows:

1. The OVERALL score shows more proximity to the most important criterion.

2. The OVERALL score is a linear function of other three criteria.

Although the first one is a key assumption on which we depend, the other one is not vital; it is only used in proof of concept experimentation.

In order to combine trust criteria, we utilized Artificial Neural Networks (ANN). In the ANN, we used 3 inputs (ON TIME DELIVERY, PRICE and QUALITY scores) and single output (OVERALL). The number of hidden layers is selected as 8. We trained our system with 60 data (in which the OVERALL score is a linear combination of the other criteria) with a learning rate of 0.8 and momentum of 0.2. The training process is repeated 2000 times.

The obtained result shows that, ANN is able to learn user preferences with limited number of data and provide higher reputation values for the providers with preferred attributes. However the implementation is in proof of concept level and may need further improvements.

CHAPTER 5

IMPLEMENTATION DETAILS

The trust model developed in scope of the thesis has been implemented as a Web application and its functionalities have been provided as Web services. The implementation is named iSURF Transitory Collaboration Service Utility (TCSU) in context of iSURF architecture.

In this chapter the implementation of iSURF TCSU has been detailed. Section 5.1 provides information on the building blocks of the implementation. In Section 5.2, implementation details of TCSU have been discussed in detail. Section 5.3 describes the interaction of the user with the system using sequence-like diagrams.

5.1 Building Blocks in iSURF TCSU

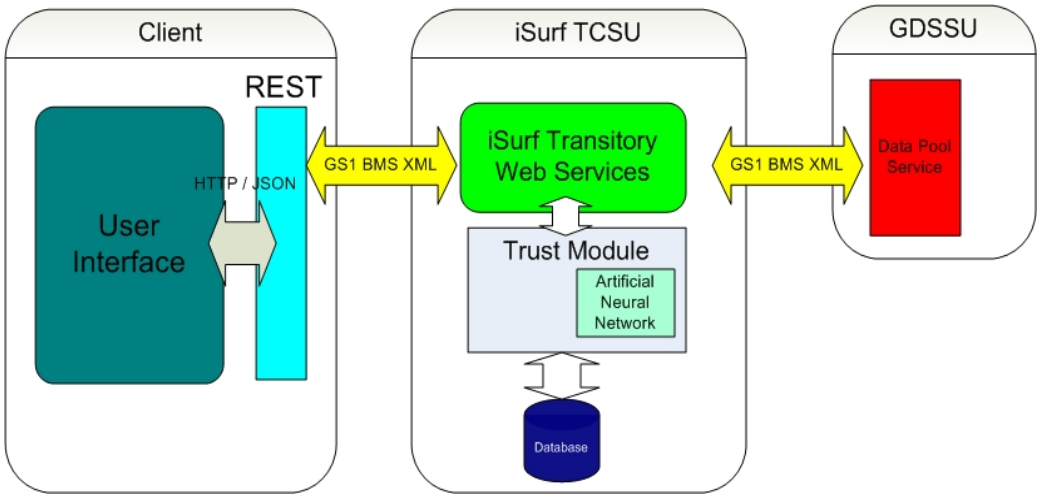

Figure 5.1- Building Blocks of iSURF TCSU

The building blocks related with iSURF TCSU can be classified into three categories according to their deployment sites. There are three main deployment locations in the system, namely: Client, TCSU and GDSSU.

On the client side, the User Interface and the RESTful service interfaces exist. iSURF TCSU, which is the main point of the architecture, hosts Transitory Web Services, Trust Module, Artificial Neural Network (ANN) Engine and the Database Server. iSurf TCSU interacts with the Data Pool Web Services which are deployed on iSURF GDSSU. The building blocks are displayed in Figure 5.1.

In the following subsections, brief information on the implemented building blocks will be provided.

- **User Interface:** In order to enable the users to interact with the system, a Web based GUI has been developed using Adobe Flex.

- **REST Services:** REST Services act as a gateway between the User Interface and TCSU. They handle simple HTTP requests from the user, create the GS1 XML documents and invoke the TCSU Web services. The outcome of the Web service calls are returned to the users as JSON objects when necessary.

- **iSURF Transitory Web Services:** iSURF Transitory Web Services handle the client requests, interact with GDSSU Web Services, invoke Trust Module operations and prepare the result documents to be returned to the client.

- **Trust Module:** Trust Module is the core component in the system that performs evaluation of the reputation values. It accesses the Database and utilizes Artificial Neural Networks.

- **Artificial Neural Networks:** In TCSU, Artificial Neural Networks are used to evaluate the scores of providers personalized for the party originating the search.

5.2 Implementation Details

In this section, implementation details of the building blocks, which have been listed in Section 5.1, are provided in a bottom up manner, i.e. the core modules of iSURF TCSU are elaborated first.

5.2.1 iSURF Trust Module

Trust module is responsible for storing, retrieving and evaluating reputation values. It is the implementation of the trust model described in Chapter 4 .Trust module follows the Data Access Object (DAO) pattern [69] in its internal structure.

5.2.1.1 Trust Model Parameters

Details of iSURF Trust model had been provided in Chapter 4. While adapting the model in TCSU, the parameters have been fixed as follows:

- **Trust Dimensions:** *Quality, Price* and *On Time Delivery* are selected as the trust criteria.
- **Valid Vote Scores:** 0,1,2,3 and 4 are valid score values. In the original model, the scores were in range [0,1].
- **Expiry Date of Votes:** Votes are effective during 18 months (t_{MAX}=18).
- **Recent Scores:** In order to determine the voting quality, scores of recent 4 months are taken into account.

5.2.1.2 Data Model

In the Trust Module, the data objects are implemented as Java Beans and persisted in the database utilizing ORM [39]techniques. Their persistence properties are annotated with JPA [40]and queried using JPQL [70] which is SQL like query language of Java Persistence. The beans are persisted and retrieved through the DAO layer, which is the only place that JPQL queries are used.

JPA abstracts the database layer and makes it independent from the RDBMS implementation. Hibernate 3.3.2GA [41] has been utilized as the JPA implementation and MySQL 5.0.67 [48] is selected as the RDBMS. Following beans are persisted in the system:

- **Consumer:** GLN of the consumer and its voting quality.
- **Provider:** GLN and basic identification of a provider
- **Vote:** One vote of a consumer to a provider, with only single dimension. Multiple dimensions are persisted as multiple votes. They also contain date information for weight decaying purposes.
- **Vote Difference:** The difference of a vote score from the evaluated score of the provider.
- **Preference Weight:** The trust scores and overall score per vote assessed by the voter. Preference weight is utilized in determining preferences by the Artificial Neural Network

In Figure 5.2, the generated database schema from the annotated Java Beans can be viewed.

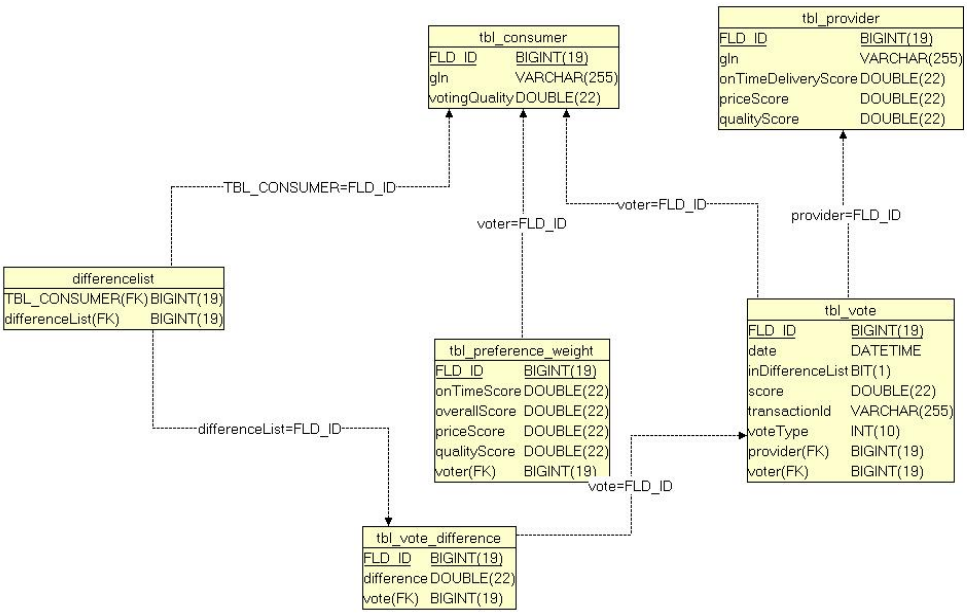

Figure 5.2- ER Diagram of the generated iSURF TCSU Database

5.2.1.3 Implementation of DAO Pattern and Dependency Injection

In conformance with the DAO pattern, usage of data sources is abstracted in a DAO Layer which is comprised of DAO classes. Access to DAO layer is performed only through a service layer (not to be confused with Web Services), which handles the business logic in the system.

In the implemented pattern, DAOs do not perform any business logic and they do not access any other DAO or Service as a convention. When a DAO method needs to access other resources and combine data, the operation is shifted to the Service Layer. On the other hand, Services can access DAOs and other Services.

In the Trust Module, the business logic is encapsulated in the operations of three services which are as follows:

- **Base Service:** It is responsible for basic persist/retrieve operations**.** Base service operations transfer their parameters to the DAO layer in order to perform basic persist/retrieve operations.

- **Trust Service:** Trust Service has the responsibility of evaluating, retrieving and updating trust values. Trust Service is the core service of iSURF TCSU that implements the reputation model. In the operations that do not require any business logic, Trust Service simply passes the parameters and delegates to the DAO layer returning the result retrieved from the DAO.

- **Preference Service:** Preference Service weights the trust values obtained through the Trust Service using an Artificial Neural Network to prioritize voting tendencies of consumers.

The services access the data objects, without needing to know how the data is queried or persisted in the data source. All operations requiring access to the data source (JPA and RDBMS in this case) are handled by DAO classes. In the Trust Module, there are three DAOs:

- **Base DAO:** Base DAO performs basic entity retrieval and persist operations.

- **Trust DAO:** Trust DAO stores and retrieves the entities (beans) related with trust and reputation.

- **Neural Network DAO:** Neural Network DAO is responsible for retrieving the entities related to multi dimensional trust criteria, to be consumed by *Preference Service* operations.

The DAOs and Services are not accessed through their actual implementations but their interfaces. This is accomplished through Dependency Injection mechanism provided by Spring Framework [36]. The actual implementation of services and DAOs are registered in the Spring Application Context and retrieved as they are requested. In this way, couplings among the implementation classes are minimized.

5.2.2 iSURF Transitory Web Services

The functionality provided by the Trust Module is exposed to the Web through Web Services. JAX-WS Reference Implementation [30] was selected as the Web Service

API for the TCSU since it conforms to JSR 224 [31] and makes development of Web services easy with its annotation support.

The Web Services have been implemented following the *contract last* approach, i.e. Java implementation of Web services are coded first and then the WSDLs are generated from the Java implementation classes. The reason behind this approach is the reality that the Trust Module is implemented first and the Web services are used to expose already existing functionality.

Web Services accomplish the following responsibilities in Transitory Collaboration Service Utility:

- Receiving and processing GS1 XML Standard Business Documents
- Calling the necessary Trust Module services to perform party registration, partner assessment and reputation evaluation.
- Integration with iSURF GDSSU
- Providing the appropriate standardized output to the clients.

There are two Web services implemented in the system. They provide similar functionalities however their invocation parameters are different. The implemented Web services are:

- Transitory Web Service (Transitory WS)
- GS1 Transitory Web Service (GS1 Transitory WS)

Transitory WS is a simple Web service invoked with the primitive Java Types. One can register a consumer or provider to TCSU, or get the combined reputation of a provider providing the Global Location Number (GLN) as a String. This service is inferior and does not expose the full capability of TCSU. It is implemented to provide an easy access to the some functionalities of TCSU using primitive data types.

GS1 Transitory WS is the main Web service exposing the whole functionality of the system to the Web. GS1 Transitory WS provides its operations aiming interoperability and conformity with GS1 standards. Thus all of the operations expect UN/CEFACT Standard Business Documents (SBD) [7] in which GS1 XML Documents are wrapped. In addition to this, the only return type is also SBD.

The operations provided by the GS1 Transitory Web Service are briefly explained below. The WSDL file of the GS1 Transitory Web Service is available in APPENDIX A.

- **registerProvider**: This operation expects a Standard Business Document wrapping a GS1 *Basic Party Registration* document. It registers the party specified in the document as a provider in the TCSU extracting its GLN from the document.

- **registerConsumer:** This operation expects a Standard Business Document wrapping a GS1 *Basic Party Registration* document. It registers the party specified in the document as a consumer in the TCSU extracting its GLN from the document.

- **vote:** This service is user to record vote of a consumer about a voter. The expected document by the operation is a *Basic Party Registration* document with some extensions. The identifications of the voter and the provider are embedded in the standard parts of *Basic Party Registration*. However voting is not included in GS1 XML documents. Therefore a custom voting document is attached to the Extensions element of *Basic Party Registration* to include the votes details of which are given in Section 5.2.4.1.

- **search:** This operation is responsible from the partner search, returning the party results along with their reputations. The service expects a Standard Business Document wrapping a GS1 Search Request from the client**.** In the Web service, the Search Request is forwarded to the GDSSU and the eligible partners are received within *GS1 Search Result* document. The parties included in the search results are evaluated through the Trust Module and

their reputation results are attached as an extension to the Search Result Document. Thereafter the search result with the reputation values is returned to the client.

5.2.3 Integration with GDSSU

In the iSURF project, Global Data Synchronization is achieved by the GDSSU. There are a number of common aspects between TCSU and GDSSU that require the integration of the two components. The areas in which the two systems need to collaborate are:

- **Registration of a Party**: Party Registration is performed by GDSSU including a user interface. The registered parties are also consumers and providers in the TCSU. Therefore GDSSU and TCSU have to synchronize the parties. On this purpose, *registerProvider* and *registerConsumer* operations are added to the GS1 Transitory Services. So within the GDSSU Party registration process, parties can be registered to the TCSU also.

- **Partner Search:** In iSURF Architecture, Partner Search is provided by GDSSU; however TCSU is the responsible block for calculation and maintenance of reputation values of the partners. Therefore integration between these two components is necessary. In order to achieve integration, Partner Search is performed through TCSU, and GDSSU Web Services are invoked by TCSU.

In the system, client sends its Search Request to the TCSU. Thereafter TCSU transforms the Search Request document through an adapter implemented for the Data Pool Service of GDSSU. The adapter serializes and calls the *globalSearch* operation of the Data Pool Web Service and unmarshalls the received Search Result to Standard Business Documents for its further processing. The final form of the Search Result is formed by TCSU.

5.2.4 Exchanged Documents

5.2.4.1 GS1 XML Documents and Extensions

Below the exchanged GS1 XML Documents in the implementation and their usage are given:

- **Basic Party Registration:** Basic Party Registration document is consumed by *registerProvider*, *registerConsumer* and *vote* operations of the GS1 Transitory Web Service. In the first two operations, GS1 documents are used without needing any extension. The GLN of the registered party is extracted from the *Information Provider* element of the Party Registration Document.

  However the standard document is extended for vote operation since it was not possible to cover the vote information in the current document. Thus a complex type named Transitory Vote is defined and the vote information is included in this complex type. A Transitory Vote element is attached to the Extensions of the gdsn:basicPartyRegistration element in the document.

- **Search Request:** Search Request is used for the partner search. The document is transmitted to the GDSSU without any modification therefore an extension is not necessary for TCSU. The GLN of the searcher is extracted from the *searchRequestIdentification* element of the Search Request.

- **Search Result:** Search Result documents are received from the GDSSU document. TCSU extracts the party identifications from the Search Result and evaluates the reputation of each party. Thereafter the reputation results are included in a complex extension type named *Transitory Score* and added to the extension section of *gdsn:searchResult*.

The extension types to the standard GS1 XML documents can be found in the XML Schema file provided in APPENDIX B.

5.2.4.2 JSON Objects

In order to avoid XML processing in the client side, the user interface is integrated using REST services. Simple HTTP GET/ POST methods and text responses are sufficient for most of the cases. However, in Partner Search, search results contain a list of providers along with their reputation values which is hard to parse as text. Therefore, REST Service wraps the Search Results in JSON before returning the response to the client. An example JSON object can be viewed in Figure 5.3.


```
[
 {
  "name": "provider2",
  "votes": {
   "ontime": 0.9942333102226257,
   "overall": 1.9964202642440796,
   "price": 3.9942333698272705,
   "quality": 2.9942333698272705
  }
 },
 {
  "name": "provider1",
  "votes": {
   "ontime": 3.9915263652801514,
   "overall": 2.9993093013763428,
   "price": 0.9915263056755066,
   "quality": 2.9915263652801514
  }
 },
]
```


Figure 5.3- Example Search Result as JSON

5.2.5 Front End

In order to realize user interaction with the TCSU, a Web based graphical user interface has been implemented using Adobe Flex.[44] Flex enables creation of rich internet applications by spending less effort.

The user interface requirements of iSURF TCSU are not complicated, i.e. there are two screens which do not have sophisticated input and outputs. One of these screens

is “Partner Assessment” screen, through which customers send their assessments on the providers. In this screen, the user puts the GLN of the partner to be assessed and provides the scores for QUALITY, PRICE and ON TIME delivery. In addition to this, user gives an overall score for the partner.

On the other screen, the users can search the providers of a product providing its Global Product Classification (GPC) code. The relevant partners are returned to the user within a Data Grid. User can select one of the partners and proceed to further negotiation steps that are explained in the following sections.

The screenshots from the two screens can be viewed in Figure 5.4 and Figure 5.5.

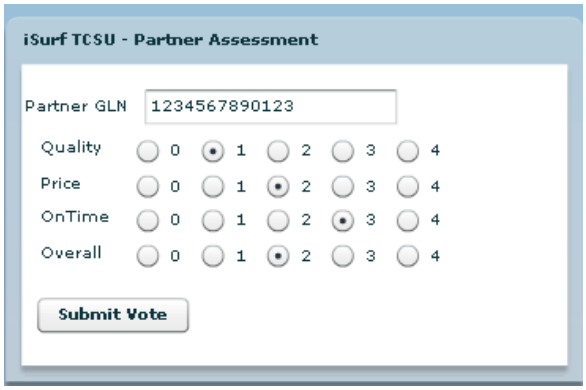

Figure 5.4 - Partner Assessment Screen

Figure 5.5 - Search Screen

The user front end is connected to the REST services through HTTP request and responses. REST Services provide the connection of the user interface to iSURF TCSU.

5.3 Application Flow

In this section, application flow for TCSU is described. TCSU is the main actor in the flow of two interactions: *Partner Assessment and Partner Search.* In the following sections, an informal description of the event flow for these interactions is given.

5.3.1 Partner Assessment

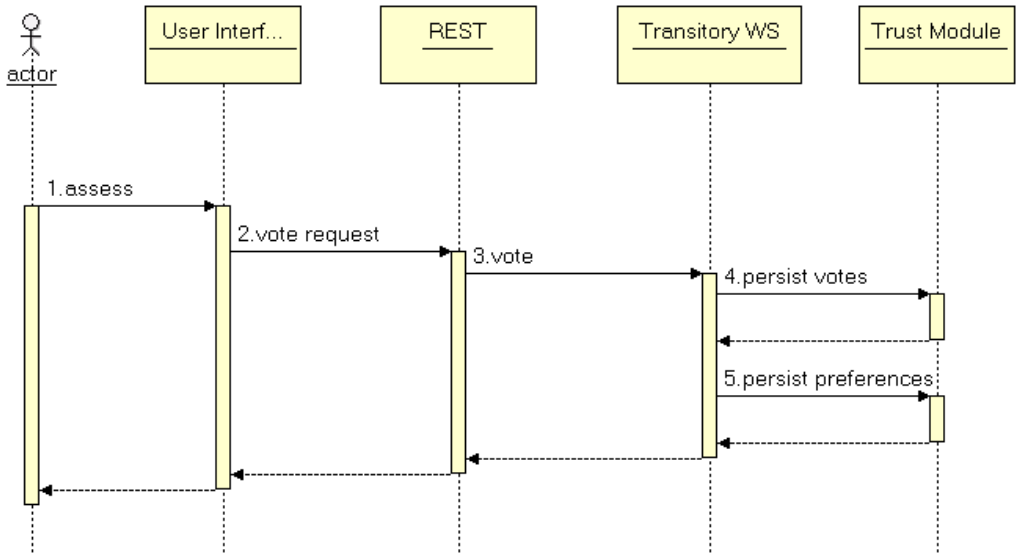

Figure 5.6- Sequence-like diagram for Partner Assessment

1. Client (consumer) selects the provider that he wants to evaluate through the Web Interface.

2. Client inputs scores for ON TIME DELIVERY, QUALITY and PRICE criteria and his OVERALL score for the provider.

3. Client sends his assessment (vote) pressing the *Submit* Button. His vote is sent as a POST request to the REST service.

4. The REST service receives the POST request and constructs the Basic *Party Registration* document with Transitory Vote extensions.

5. The REST Service invokes the vote operation of the GS1 Transitory Web Service.

6. GS1 Transitory Web Service extracts the voter GLN and Votes (votes for each criterion) from the received Basic Party Registration document.

7. GS1 Transitory Web Service determines the user priority among criteria from the Votes.

8. GS1 Transitory Web Service stores the votes and user priority.

9. Interaction is completed.

5.3.2 Partner Search

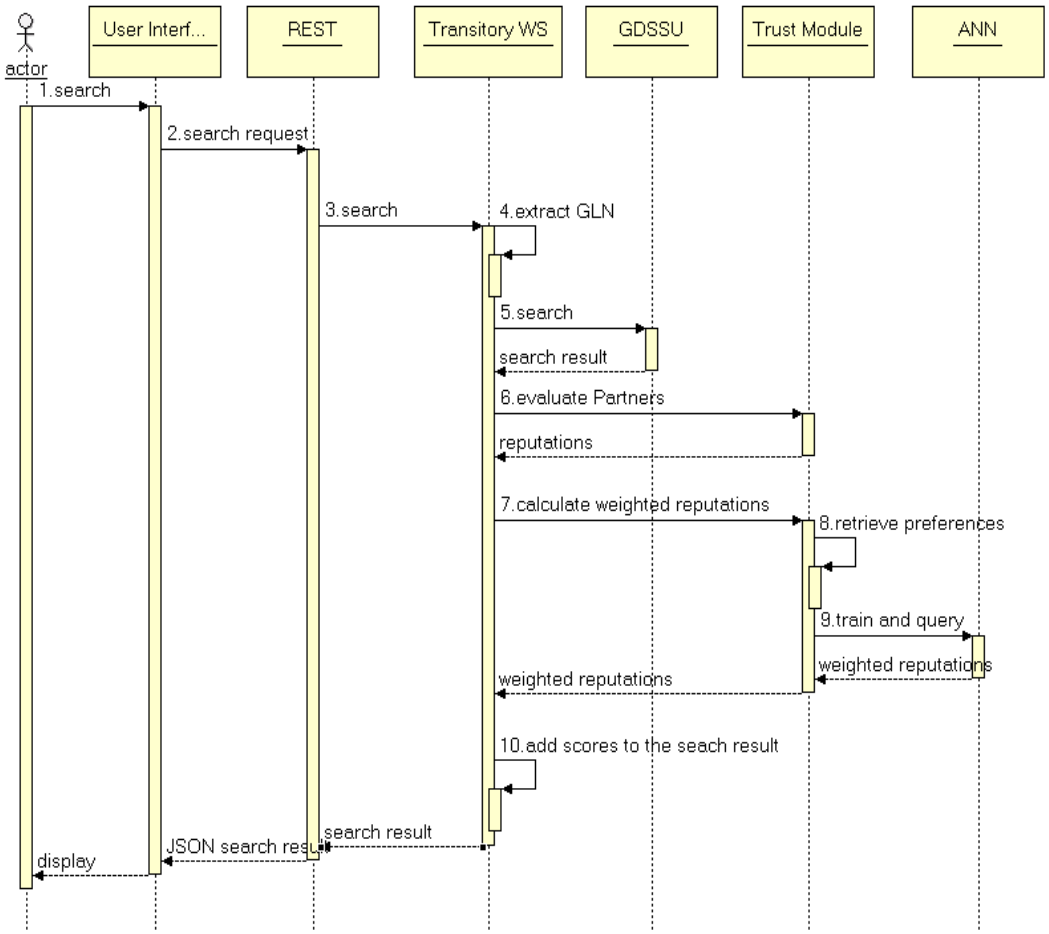

Figure 5.7 - Sequence-like Diagram for Partner Search

1. Client inputs the classification code (GPC) of the product, the providers of which he wants to retrieve.

2. Client presses Search Button to initiate the Partner Search. The GPC is sent to the REST service as a HTTP GET request.

3. REST Service receives the request, creates a Search Request Document with the GPC provided by the client and invokes *search* operation of TCSU GS1 Transitory Web Service.

4. GS1 Transitory Web Service extracts the GLN of the client from the Search Request.

5. GS1 Transitory Web Service extracts the GPC from the Search Request and invokes globalSearch operation of GDSSU Data Pool Web Service.

6. GDSSU Data Pool Web Service returns a Search Result to GS1 Transitory Web Service containing the information of the parties providing the trade item classified with the GPC.

7. GS1 Transitory Web Service extracts the returned party identification from the Search Result.

8. GS1 Transitory Web Service evaluates each of the parties generating scores for ON TIME DELIVERY, QUALITY and PRICE.

9. GS1 Transitory Web Service determines overall score for the parties taking the past preferences of the client (originator of the search) through Artificial Neural Networks.

10. GS1 Transitory Web Service attaches the multi-criteria scores and overall reputation within a Transitory Score element in the Search result and returns to the REST Service.

11. REST Service extracts the Transitory Scores and party information, packages them into a JSON array and writes the array to its HTTP response.

12. On the user side the search result in JSON array is formatted and displayed to the user.

CHAPTER 6

CONCLUSION AND FUTURE WORK

In scope of this thesis work, we have developed a reliable and robust trust and reputation management system for partner search, in order to establish transitory collaborations for the exceptional cases occurring in supply chains.

Development of the reputation system has been conducted in two phases. Firstly, previous studies related with trust and reputation have been reviewed. In the review, we have identified that a number of realistic reputation model requirements have not been addressed in the previous work. These requirements are mainly handling of non-ideal voter behavior and multiple-reputation criteria. Based on the previous studies, we have developed a reputation model which also addresses these two requirements. We have included the voting quality of the voters in reputation computation to manage behavior of non-ideal voters. In order to aggregate multiple reputation criteria in a personalized manner for the searching party, artificial neural networks have been utilized. Experimental results show that, the developed model provides a robust reputation management mechanism.

Thereafter the reputation model has been implemented within an application, in context of iSURF Project. The reputation data model has been transferred to a database platform and the partner search and partner assessment functionalities are exposed as Web services. In order to ease user interaction with the system, the client side has been empowered with REST services and a Web based graphical user interface. For the interaction among the entities in the implementation, GS1 XML messages are utilized and extended to cover reputation related information.

As the result of this thesis, a reliable reputation management system has been developed and implemented.

We have strived to meet the requirements of a realistic trust and management system in this study; however there exist a number of issues that can be addressed based on the thesis work in order to realize a more reliable and enhanced reputation mechanism.

First of all, aggregation of multiple reputation criteria constitutes a significant part for the future work. In the thesis, we have implemented a proof of concept aggregation methodology using Artificial Neural Networks. In order to obtain a more concrete fusion of the criteria with neural networks, the approach can be enhanced taking the available reputation criteria in the targeted domains into consideration. In addition to this, the stability of the neural networks against noisy data from the voters can be improved.

In our system, the partner assessment is performed by the consumers manually. This yields to a subjective assessment, which can be also error prone. However some of the trust criteria such as price and delivery latency can be obtained automatically from existing electronic business mechanisms such as business or planning processes. A further study can be incorporation of these kinds of data in the reputation managements systems, in order to provide a move objective assessment for the partners.

REFERENCES

[1] iSURF Project Website, http://www.srdc.com.tr/isurf/, last visited on June 2009

[2] EU Seventh Framework Programme (FP7), http://cordis.europa.eu/fp7/ , last visited on June 2009

[3] GS1 Home, http://www.gs1.org/, last visited on June 2009

[4] OASIS Universal Business Language (UBL), http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ubl, last visited on June 2009

[5] OASIS: Advancing open standards for the global information society, http://www.oasis-open.org/home/index.php, last visited on June 2009

[6] Collaborative Planning, Forecasting & Replenishment (CPFR®), http://www.vics.org/committees/cpfr/ , last visited on June 2009

[7] GS1 XML, http://www.gs1.org/ecom/xml, last visited on June 2009

[8] GS1 GDSN® is the GS1 Global Data Synchronisation Network, http://www.gs1.org/gdsn, last visited on June 2009

[9] Uniform Code Council (UCC), http://www.uc-council.org/, last visited on June 2009

[10] Electronic Commerce Council of Canada (ECCC), http://www.gs1ca.org/home.asp, last visited on June 2009

[11] European Article Numbering-Uniform Code Council, http://en.wikipedia.org/wiki/European_Article_NumberingUniform_Code_Council, last visited on June 2009

[12] GS1 Türkiye, http://gs1.tobb.org.tr/, last visited on June 2009

[13] GS1 EANCOM, http://www.gs1.org/ecom/eancom, last visited on June 2009

[14] GTIN: Global Trade Identification Number Information, http://www.gtin.info, last visited on June 2009

[15] What is Global Product Classification (GPC)?, http://www.gs1.org/gdsn/gpc, last visited on June 2009

[16] Global Location Numbers (GLN), http://www.gs1.org/docs/idkeys/GS1_Global_Location_Numbers.pdf, last visited on June 2009

[17] GS1 eCom, http://www.gs1.org/ecom, last visited on June 2009

[18] Electronic Data Exchange(EDI), http://en.wikipedia.org/wiki/Electronic_Data_Interchange, last visited on June 2009

[19] GS1 Global Data Dictionary, http://gdd.gs1.org/gdd/public/default.asp, last visited on June 2009

[20] United Nations Centre for Trade Facilitation and Electronic Business, http://www.unece.org/cefact/, last visited on June 2009

[21] UN/CEFACT Core Components Technical Specification version 3, http://ebxml.xml.org/node/163, last visited on June 2009

[22] AS1 standards info, http://www.drummondgroup.com/html-v2/as1.html, last visited on June 2009

[23] AS2 (Application Statement 2) Technical Specifications, http://www.ietf.org/rfc/rfc4130.txt, last visited on June 2009

[24] Drummond Group Inc. (DGI) Homepage, http://www.drummondgroup.com, last visited on June 2009

[25] METU - Software Research and Development Center, http://www.srdc.metu.edu.tr, last visited on June 2009

[26] W3C Web Services Glossary, http://www.w3.org/TR/ws-gloss/#webservice, last visited on June 2009

[27] SOAP Specifications, http://www.w3.org/TR/soap/, last visited on June 2009

[28] Web Services Description Language (WSDL) 1.1, http://www.w3.org/TR/wsdl, last visited on June 2009

[29] UDDI :Online community for the Universal Description, Discovery, and Integration, http://uddi.xml.org/, last visited on June 2009

[30] JAX-WS-Java API for XML Web Services and Reference Implementation, https://jax-ws.dev.java.net/, last visited on June 2009

[31] JSR 224: JavaTM API for XML-Based Web Services (JAX-WS) 2.0, http://jcp.org/en/jsr/detail?id=224, last visited on June 2009

[32] Building Web Services the REST Way, http://www.xfront.com/REST-Web-Services.html, last visited on June 2009

[33] REST:: Representational State Transfer, http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm, last visited on June 2009

[34] JSON, http://www.json.org/, last visited on June 2009

[35] JAXB -Java Architecture for XML Binding, https://jaxb.dev.java.net/, last visited on June 2009

[36] About Spring , http://www.springsource.org/about, last visited on June 2009

[37] Enterprise JavaBeans Technology, http://java.sun.com/products/ejb, last visited on June 2009

[38] Inversion of control, http://en.wikipedia.org/wiki/Inversion_of_control, last visited on June 2009

[39] ORM - Object-relational mapping, http://en.wikipedia.org/wiki/Object-relational_mapping, last visited on June 2009

[40] The Java Persistence API - A Simpler Programming Model for Entity Persistence, http://java.sun.com/developer/technicalArticles/J2EE/jpa/, last visited on June 2009

[41] Hibernate: Relational Persistence for Java and .NET, https://www.hibernate.org/, last visited on June 2009

[42] JSR 220: Enterprise JavaBeans™ 3.0, http://jcp.org/en/jsr/detail?id=220, last visited on June 2009

[43] Adobe Flex, http://www.adobe.com/products/flex/, last visited on June 2009

[44] Adobe Flash, http://www.adobe.com/products/flash/, last visited on June 2009

[45] MXML, http://en.wikipedia.org/wiki/MXML, last visited on June 2009

[46] ActionScript, Home Page, http://www.actionscript.org/, last visited on June 2009

[47] Joone - Java Object Oriented Neural Engine, www.jooneworld.com, last visited on June 2009

[48] MySQL Website , http://www.mysql.com/, last visited on June 2009

[49] Uses of MySQL, http://en.wikipedia.org/wiki/MySQL#Uses, last visited on June 2009

[50] Apache Tomcat Homepage, http://tomcat.apache.org/, last visited on June 2009

[51] Java Servlet Technology, http://java.sun.com/products/servlet/, last visited on June 2009

[52] JavaServer Pages Technology, http://java.sun.com/products/jsp/, last visited on June 2009

[53] eBay, http://www.ebay.com/ , last visited on June 2009

[54] Amazon Auctions, http://auctions.amazon.com/, last visited on June 2009

[55] onSale.com, http://www.onsale.com , last visited on June 2009

[56] Giorgos Zacharia, Alexandros Moukas, Pattie Maes: Collaborative reputation mechanisms for electronic marketplaces. Decision Support Systems 29(4): 371-388 (2000)

[57] Jordi Sabater, Carles Sierra: REGRET: reputation in gregarious societies. Agents 2001: 194-195

[58] Arturo Avila-Rosas, Michael Luck: A Direct Reputation Model for VO Formation. CEEMAS 2005: 460-469

[59] ECOLEAD Project Web Site, http://ecolead.vtt.fi/, last visited on June 2009

[60] Bin Yu and Munindar P. Singh, A Social Mechanism for Reputation Management in Electronic Communities (pdf) In Proceedings of the 4th International Workshop on Cooperative Information Agents (CIA) Springer-Verlag, Berlin, 2000.

[61] Mudhakar Srivatsa, Li Xiong, Ling Liu: TrustGuard: countering vulnerabilities in reputation management for decentralized overlay networks. WWW 2005: 422-431

[62] Audun Josang, The Beta Reputation System, 15th Bled Electronic Commerce Conference, e-Reality: Constructing the e-Economy Bled, Slovenia, June 17 - 19, 2002

[63] Beta Distribution, http://en.wikipedia.org/wiki/Beta_distribution, last visited June 2009

[64] Zhengqiang Liang, Weisong Shi: Performance evaluation of rating aggregation algorithms in reputation systems. CollaborateCom, 2005

[65] Definition of Trust in Webster, http://www.merriam-webster.com/dictionary/trust, last visited on June 2009

[66] ECOLEAD Deliverable D21.4b – Creating and Supporting Trust Culture in VBEs, http://www.ve-forum.org/projects/284/Deliverables/D21.4b_Final.pdf, last visited on June 2009

[67] Gambetta, Diego (ed.), Can we trust trust? Trust: Making and Breaking Cooperative Relations

[68] Jordi Sabater, Carles Sierra: Review on Computational Trust and Reputation Models. Artif. Intell. Rev. 24(1): 33-60 (2005)

[69] Data access object, http://en.wikipedia.org/wiki/Data_Access_Object, last visited on June 2009

[70] JPQL - The Java Persistence Query Language, http://java.sun.com/javaee/5/docs/tutorial/backup/update3/doc/QueryLanguage.html, last visited on June 2009

APPENDIX A

WSDL FOR GS1 TRANSITORY WEB SERVICE

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://ws.transitory.isurf.srdc.com.tr/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/"
name="GS1TransitoryWS" targetNamespace="http://ws.transitory.isurf.srdc.com.tr/">
   <types>
      <xsd:schema>
         <xsd:import namespace="urn:ean.ucc:gdsn:2"
schemaLocation="http://localhost:8080/isurf-transitory-ws/gs1-transitory-ws?xsd=1"/>
      </xsd:schema>
      <xsd:schema>
         <xsd:import namespace="urn:ean.ucc:align:2"
schemaLocation="http://localhost:8080/isurf-transitory-ws/gs1-transitory-ws?xsd=2"/>
      </xsd:schema>
      <xsd:schema>
         <xsd:import namespace="urn:ean.ucc:2"
schemaLocation="http://localhost:8080/isurf-transitory-ws/gs1-transitory-ws?xsd=3"/>
      </xsd:schema>
      <xsd:schema>
         <xsd:import
namespace="http://www.unece.org/cefact/namespaces/StandardBusinessDocumentHeader"
schemaLocation="http://localhost:8080/isurf-transitory-ws/gs1-transitory-ws?xsd=4"/>
      </xsd:schema>
      <xsd:schema>
         <xsd:import namespace="http://ws.transitory.isurf.srdc.com.tr/"
schemaLocation="http://localhost:8080/isurf-transitory-ws/gs1-transitory-ws?xsd=5"/>
      </xsd:schema>
      <xsd:schema>
         <xsd:import namespace="http://isurf.srdc.com.tr/Transitory"
schemaLocation="http://localhost:8080/isurf-transitory-ws/gs1-transitory-ws?xsd=6"/>
      </xsd:schema>
   </types>
   <message name="search">
      <part name="parameters" element="tns:search"/>
   </message>
   <message name="searchResponse">
      <part name="parameters" element="tns:searchResponse"/>
   </message>
   <message name="registerVoter">
      <part name="parameters" element="tns:registerVoter"/>
   </message>
   <message name="registerVoterResponse">
      <part name="parameters" element="tns:registerVoterResponse"/>
   </message>
   <message name="registerProvider">
      <part name="parameters" element="tns:registerProvider"/>
   </message>
   <message name="registerProviderResponse">
      <part name="parameters" element="tns:registerProviderResponse"/>
   </message>
   <message name="vote">
      <part name="parameters" element="tns:vote"/>
   </message>
   <message name="voteResponse">
```

```
      <part name="parameters" element="tns:voteResponse"/>
   </message>
   <portType name="GS1TransitoryWS">
      <operation name="search">
         <input message="tns:search"/>
         <output message="tns:searchResponse"/>
      </operation>
      <operation name="registerVoter">
         <input message="tns:registerVoter"/>
         <output message="tns:registerVoterResponse"/>
      </operation>
      <operation name="registerProvider">
         <input message="tns:registerProvider"/>
         <output message="tns:registerProviderResponse"/>
      </operation>
      <operation name="vote">
         <input message="tns:vote"/>
         <output message="tns:voteResponse"/>
      </operation>
   </portType>
   <binding name="GS1TransitoryWSPortBinding" type="tns:GS1TransitoryWS">
      <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
      <operation name="search">
         <soap:operation soapAction=""/>
         <input>
            <soap:body use="literal"/>
         </input>
         <output>
            <soap:body use="literal"/>
         </output>
      </operation>
      <operation name="registerVoter">
         <soap:operation soapAction=""/>
         <input>
            <soap:body use="literal"/>
         </input>
         <output>
            <soap:body use="literal"/>
         </output>
      </operation>
      <operation name="registerProvider">
         <soap:operation soapAction=""/>
         <input>
            <soap:body use="literal"/>
         </input>
         <output>
            <soap:body use="literal"/>
         </output>
      </operation>
      <operation name="vote">
         <soap:operation soapAction=""/>
         <input>
            <soap:body use="literal"/>
         </input>
         <output>
            <soap:body use="literal"/>
         </output>
      </operation>
   </binding>
   <service name="GS1TransitoryWS">
      <port name="GS1TransitoryWSPort" binding="tns:GS1TransitoryWSPortBinding">
         <soap:address location="http://localhost:8080/isurf-transitory-ws/gs1-
transitory-ws"/>
      </port>
   </service>
</definitions>
```

APPENDIX B

XML SCHEMAS OF THE GS1 EXTENSIONS FOR TRUST AND REPUTATION DATA

```
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:isurf="http://isurf.srdc.com.tr/Transitory"
targetNamespace="http://isurf.srdc.com.tr/Transitory" elementFormDefault="qualified">
   <simpleType name="Provider">
      <restriction base="string"/>
   </simpleType>
   <complexType name="TransitoryVote">
      <sequence>
         <element name="provider" type="isurf:Provider"/>
         <element name="votes" type="isurf:Votes"/>
      </sequence>
   </complexType>
   <complexType name="Votes">
      <sequence>
         <element name="qualityVote" type="isurf:Vote"/>
         <element name="onTimeVote" type="isurf:Vote"/>
         <element name="priceVote" type="isurf:Vote"/>
         <element name="overallVote" type="isurf:Vote"/>
      </sequence>
   </complexType>
   <simpleType name="Vote">
      <restriction base="int">
         <minInclusive value="0"/>
         <maxInclusive value="4"/>
      </restriction>
   </simpleType>
   <element name="TransitoryVote" type="isurf:TransitoryVote"/>
   <element name="TransitoryScore" type="isurf:TransitoryScore"/>
   <complexType name="TransitoryScore">
      <sequence>
         <element name="Provider" type="isurf:Provider"/>
         <element name="Scores" type="isurf:Scores"/>
      </sequence>
   </complexType>
   <complexType name="Scores">
      <sequence>
         <element name="qualityScore" type="isurf:Score"/>
         <element name="onTimeScore" type="isurf:Score"/>
         <element name="priceScore" type="isurf:Score"/>
         <element name="overallScore" type="isurf:Score"/>
      </sequence>
   </complexType>
   <simpleType name="Score">
      <restriction base="float">
         <minInclusive value="0"/>
         <maxInclusive value="4"/>
      </restriction>
   </simpleType>
</schema>
```