


TRUST MANAGEMENT WITH CONFIDENTIALITY FOR DATA SHARING,
ERBAC FOR SECURE DATA WAREHOUSE, ERP

APPROVED BY SUPERVISORY COMMITTEE:


Dr. Bhavani Thuraisingham, Chair


Dr. Latifur Khan


Dr. Murat Kantarcioglu

Copyright 2007

Srinivasan Iyer

All Rights Reserved

TRUST MANAGEMENT WITH CONFIDENTIALITY FOR DATA SHARING,
ERBAC FOR SECURE DATA WAREHOUSE, ERP

by

SRINIVASAN IYER, B.E.

THESIS

Presented to the Faculty of
The University of Texas at Dallas

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN
COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT DALLAS

May 2007

UMI Number: 1441877

UMI[®]

UMI Microform 1441877

Copyright 2007 by ProQuest Information and Learning Company.
All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

PREFACE

This thesis was produced in accordance with guidelines which permit the inclusion as part of the thesis the text of an original paper or papers submitted for publication. The thesis must still conform to all other requirements explained in the "Guide for the Preparation of Master's Theses and Doctoral Dissertations at The University of Texas at Dallas." It must include a comprehensive abstract, a full introduction and literature review and a final overall conclusion. Additional material (procedural and design data as well as descriptions of equipment) must be provided in sufficient detail to allow a clear and precise judgment to be made of the importance and originality of the research reported.

It is acceptable for this thesis to include as chapters authentic copies of papers already published, provided these meet type size, margin and legibility requirements. In such cases, connecting texts which provide logical bridges between different manuscripts are mandatory. Where the student is not the sole author of a manuscript, the student is required to make an explicit statement in the introductory material to that manuscript describing the student's contribution to the work and acknowledging the contribution of the other author(s). The signatures of the Supervising Committee which precede all other material in the thesis attest to the accuracy of this statement.

ACKNOWLEDGEMENTS

I thank my advisor Dr. Bhavani Thuraisingham for giving me an opportunity to work on this research project and for her constant guidance, support and motivation throughout this research work. I also thank Dr. Murat Kantarcioglu and Dr. Latifur Khan for being a part of the committee and for their kind co-operation.

I also thank AFOSR and Texas Enterprise Funds for their support and fund for the project “Information Operation across Info spheres”. My first part of the research is a part of the research design and simulation of trust management techniques for a coalition data sharing environment was a part of the AFOSR project.

I thank my peers in the Data and Application Security Group for their constant support and feedback until the completion of this research project.

April 2007

TRUST MANAGEMENT WITH CONFIDENTIALITY FOR DATA SHARING,
ERBAC FOR SECURE DATA WAREHOUSE, ERP

Publication No. _____

Srinivasan Iyer, M.S.
The University of Texas at Dallas, 2007

Supervising Professor: Dr. Bhavani Thuraisingham

My thesis consists of two different research topics. My first part of the thesis deals with the development of trust management techniques without disclosing confidential information. The objective of the project is to simulate a peer to peer communication network in which each system acts as a software agent. Each agent as a set of information with it and the aim of the simulation is to gather as much as information possible from the other agents in the network co-existing in the same session. The agents start to collect the information related to that of they already have in their database. The information has three things linked with it. Data, Token number of the message which is in sequential order provides more information, Domain level of the data which shows the access level of the data. There can be multiple copies of the information in the network. The combination of data and token number for any information is unique. The domain levels various in different users. They collect and share information through neighbors and maintain a trust table for the all the agents in the session. The trust level is stored in the history, if the same agent comes in contact in the future session the trust level from the previous sessions are loaded. The agents increase the trust additively for any new or correct information, if they find any discrepancy in the information which can be found by the multiple copies existing in different agents or the mismatch in the token number and data combination the trust level of that corresponding agent is reduced. This

goes on until the session ends or any agent collects all information required. The end of the simulation gives the amount of information gained and lost in each session.

The second part of the research deals with finding the issues in designing and building a secure data warehouse. The research clearly finds out the security issues, discusses the technologies available in designing secure data warehouses, the design steps involved and their issues and advantages. The main objective of the research is to design a security component for data warehouse. The design phase discusses the existing Role based access control techniques which are used by most of the existing data management systems. Finds the issues in RBAC, Defines a unified framework which unites RBAC and usage control (UCON) proposed as Extended Role based access controls. The research includes simulation of an inventory system with ERBAC implemented which show cases all possible advantages. The second part of the project also includes a research in implementing the above proposed ERBAC into an Enterprise wide system which is similar to that of a data warehouse. The project also illustrates the advantages of the ERBAC component in an Enterprise wide system.

TABLE OF CONTENTS

Preface.....	iv
Acknowledgements.....	v
Abstract.....	vi
List of Figures.....	xi
List of Charts.....	xii
CHAPTER 1 INTRODUCTION	1
CHAPTER 2 TRUST MANAGEMENT WITH CONFIDENTIALITY TECHNIQUES FOR A COALITION DATA SHARING ENVIRONMENT.....	4
2.1 Abstract.....	5
2.2 Introduction.....	5
2.3 Preliminaries	7
2.3.1 Definitions.....	7
2.3.2 Functions.....	8
2.4 Background and Related Work.....	8
2.5 System Architecture.....	9
2.6 Implementation of the System	12
2.6.1 Overview.....	12
2.6.2 Algorithm for Information sharing with Confidentiality and Trustworthy Computing.....	13
2.6.3 Specifications of Algorithm.....	14
2.7 Experimental Results	15
2.8 Summary and Directions.....	16
References.....	18
CHAPTER 3 EXTENDED RBAC –BASED DESIGN AND IMPLEMENTATION FOR A SECURE DATAWAREHOUSE	20

3.1	Abstract	21
3.2	Overview	21
3.3	Security for Data Warehouses	22
3.3.1	Security Issues	22
3.3.2	Technologies for Secure Warehousing	23
3.3.3	Design Steps.....	27
3.4	Design of Extended RBAC for Secure Data Warehousing	30
3.4.1	Overview	30
3.4.2	Role Based Access Control for Data Warehousing	31
3.4.2.1	Overview	31
3.4.2.2	Issues with the Existing Models	32
3.4.3	ERBAC System Architecture for Data Warehouses.....	32
3.4.3.1	Overview of ERBAC	32
3.4.3.2	System Architecture of ERBAC	33
3.4.3.3	Implementation of ERBAC in Data Warehousing.....	34
3.4.3.4	Experiment and Challenges	36
3.5	Summary and Directions.....	37
	References.....	39
 CHAPTER 4 ERBAC IN ENTERPRISE RESOURCE PLANNING SYSTEMS.....		40
4.1	Abstract	41
4.2	Introduction.....	41
4.2.1	Enterprise Wide Systems (ERP) Overview	41
4.3	History and Background of ERP.....	42
4.3.1	History of ERP	42
4.3.2	Background of RBAC in ERP	43
4.4	ERP Security Issues	43
4.5	RBAC in ERP	44
4.6	General Implementation of RBAC in ERP.....	46
4.6.1	Overview.....	46
4.7	Issues of RBAC in ERP	47
4.8	UCON Overview	48
4.9	Extended Role based Access Control	49

4.9.1	ERBAC Overview	49
4.9.2	ERBAC System Architecture	49
4.10	ERBAC in ERP System Architecture	51
4.10.1	Security Based Architecture Overview.....	51
4.11	Challenges in Implementing ERBAC in ERP	53
	References.....	55
CHAPTER 5 CONCLUSION.....		57

Vita

LIST OF FIGURES

Number	Page
1. System Model for agent information sharing with confidentiality and trust.	10
2. Flow diagram of the Information sharing with confidentiality and trust.	13
3. Secure Data Warehouse Example.	23
4. Security Policy Integration.	24
5. Security Policy Integration.	28
6. ERBAC Architecture with UCONABC Extension.	34
7. ERBAC in Enterprise Wide Data Warehouse.	35
8. RBAC Component implemented in ERP Application.	45
9. UCONABC Model Component.	48
10. ERBAC System Architecture in ERP System.	52

LIST OF CHARTS

Number	Page
1. Net gain of information by each agent in four continuous sessions of information sharing with trust computing.	15

CHAPTER 1
INTRODUCTION

Srinivasan Iyer

The Department of Computer Science

The University of Texas at Dallas

P.O. Box 830688

Richardson, Texas 75083-0688

This documentation mainly elaborates about my research during my graduate studies. The research comes under the Data Applications and its Security Issues. It consists of three different topics under the Application security. The first part of the thesis mainly focuses on coalition data sharing environment. The project was partially funded by US Air force under the project “Information operations across info spheres”. The project consists of various aspects like Data Mining in a coalition environment, Trust management between the organizations. In the coalition data sharing environment Trust management plays an important role, If not there is a chance of losing more data from each organization.

My part of this project was to develop a trust management technique with confidentiality among peers. The objective of the project is to simulate a peer to peer communication network in which each system acts as a software agent. Each agent as a set of information with it and the aim of the simulation is to gather as much as information possible from the other agents in the network co-existing in the same session. The agents start to collect the information related to that of they already have in their database. The information has three things linked with it. Data, Token number of the message which is in sequential order provides more information, Domain level of the data which shows the access level of the data. There can be multiple copies of the information in the network. The combination of data and token number for any information is unique. The domain levels various in different users. They collect and share information through neighbors and maintain a trust table for the all the agents in the session. The trust level is stored in the history, if the same agent comes in contact in the future session the trust level from the previous sessions are loaded. The agents increase the trust additively for any new or correct information, if they find any discrepancy in the information which can be found by the multiple copies existing in different agents or the mismatch in the token number and data combination the trust level of that corresponding agent is reduced. This goes on until the session ends or any agent collects all information required. The end of the simulation gives the amount of information gained and lost in each session. This part of the project was also published in FTDCS 2007 IEEE Conference held at Sedona, Arizona.

The third and fourth chapter of my thesis involves two systems which are not much different from one another Data Warehousing and Enterprise Wide Systems. The aim of this project is to design an extended role based access control for Secure Data warehousing and

Enterprise Wide Systems. The research clearly finds out the security issues, discusses the technologies available in designing secure data warehouses, the design steps involved and their issues and advantages. The main objective of the research is to design a security component for data warehouse. The design phase discusses the existing Role based access control techniques which are used by most of the existing data management systems. Finds the issues in RBAC, Defines a unified framework which unites RBAC and usage control (UCON) proposed as Extended Role based access controls. The research includes simulation of an inventory system with ERBAC implemented which show cases all possible advantages. The second part of the project also includes a research in implementing the above proposed ERBAC into an Enterprise wide system which is similar to that of a data warehouse. The project also illustrates the advantages of the ERBAC component in an Enterprise wide system. It digs into the existing ERP system, Checks the issues in the existing Role based access controls, Discusses the advantages of Unified Framework developed, Implementation Issues. This paper was published in ARES 2007 IEEE conference Vienna, Austria. Chapter three clearly describes ERBAC for Secure Data Warehouse. Chapter four describes Secure ERP System. The Conclusion is in chapter 5 with a discussion about the future directions

CHAPTER 2
**TRUST MANAGEMENT WITH CONFIDENTIALITY TECHNIQUES FOR A
COALITION DATA SHARING ENVIRONMENT**

Srinivasan Iyer and Dr.Bhavani Thuraisingham

The Department of Computer Science

The University of Texas at Dallas

P.O. Box 830688

Richardson, Texas 75083-0688

Copyright© 2007 IEEE. Reprinted with permission from 11th IEEE International
Workshop on Future Trends of Distributed Computing Systems. FTDCS 2007

2.1 Abstract

Effective communication among agents in large teams is crucial because the members share a common goal but only have partial views of the environment. Information sharing is difficult in a large team because, a team member may have a piece of valuable information but not know who needs the information, since it is infeasible to know what each other agent is doing. Information sharing is a main part of any system or organization. The information sharing needs to be foolproof. Only the legitimate receiver should be able to get hold of the information. This chapter mainly deals with intelligent software agents for information sharing with confidentiality and trust. It clearly defines an Intelligent Software Agent, background of Information sharing in intelligent agents and the trust in the agents. Some part of the information needs confidentiality. The information that is shared requires security policy enforced based on the domain of the information and trust level of individual agent. This chapter also provides the results of a multi-agent simulation for sharing information. It also implements trust calculation based on the quality of information provided by the peer agents in the simulation.

2.2 Introduction

Information sharing is necessary and unavoidable, even in the times of Kings and Empires. There were many alliances between the kingdoms, espionages, miscommunications, treachery, deception, compromises, victories and defeats. The information sharing needs to be secure. That is, it is critical that the information does not get into the wrong hands. Only the legitimate receiver should be able to get hold of the information. Even Kings had their own way of secured Information sharing. They had the royal seals to verify if the information is authentic.

Exciting emerging applications require hundreds or thousands of agents and robots to coordinate to achieve their joint goals. In domains such as military operations, space or disaster response, coordination among large numbers of agents promises to revolutionize the effectiveness of our ability to achieve complex goals. Such domains are characterized by widely distributed entities with limited communication channels among them and no agent having a complete view of the environment. Information relevant to team goals will become available to team members in a spontaneous, unpredictable and, most importantly, distributed

way. The question addressed in this chapter is when a team member senses some information, how it can decide which team member to communicate that information to. In most applications for very large teams, broadcasting information is not suitable, desirable or feasible. Instead, the agent must attempt to target its information delivery to just the agents that need it. In a large team, each member has a limited model of what other members of the group know or even what many of them are doing. For example, a field agents involved in a military operation may observe many features of a battlefield on route to an assignment. Many of its observations will be relevant to the plans of other combatants but the field agents will not necessarily know which group members require the information.

Since 9/11, the agencies have moved to a need to know paradigm to a need to share paradigm. For many applications it is important that the information be shared and then examines the consequences. There are now efforts on information sharing based on Trust. That is, do I trust say John enough to share some critical information. What happens if I trust John only 50% of the time? Do I still share the information with him? Another good example is coalition data sharing between countries. To fight the global war on terror, organizations have to share data between trustworthy and untrustworthy as well as semi-trustworthy partners. What should say the United States do when there is a need to share data between us and a partner who we believe is untrustworthy, but is still part of our coalition to fight the global war on terror? In our previous paper [17] on Assured Information Sharing, we have discussed the various pros and cons on the need to share model for data sharing.

This chapter presents a system to sharing information that is applicable to large teams [1]. A key to the solution is imposing a static network topology on the members of the team where each agent requiring communication to be only along very few links in that network. The key observation underlying this solution is that each piece of information is interrelated and the sender of a piece of information can "guess" who might need some information based on previously sent messages. Thus, when an agent has a piece of information, it can determine which of its neighbors in the network is most likely to either need the information or know who does, based on related messages previously received. Secondly, investigate the influence of different types of team network topology on the efficiency of information sharing.

Trust negotiation is a very important part of any system or an organization. Without trust

no transaction can be successful. If there are many systems interacting between them each one has to have trust with other in order to share data, alliances and deals to save the operation cost which is major part of any project. The negotiation is always conflicting since it is to compromise between two agents in order to achieve decision for conflicting distributed systems. The negotiation is taken based on the environment with two decisions to support self interest or the entire system. The decision tree is then formed based on the negotiation and the scenario is stored into the library incase if it is newly proposed. So that it can be used in the future without much of computation.

The Confidentiality of Information is a major threat in a system that is used to share information. In case the confidential information is disclosed to an agent that is not entitled to that level of security, there is a possibility of losing the vital information to an untrustworthy agent. If the trust level of the agent does not match with the security level of the information then the information is secured. The security policy of the information is distinguished into four types as D1, D2, D3 and D4. We call them domains. For more details on security policies, we refer to [18].

2.3 Preliminaries

2.3.1 Definitions

Agent: An agent defines a person or an organization that interacts with other person or organization on behalf of the owner.

Software Agent: It is not as simple as a real world agent. There are various definitions for a software agent. The closest definition would be the following “A software agent is a software with some inbuilt functionalities that interacts with other software agents and perform the allocated task based on the rules that govern them.”

Intelligent Software Agent: It is a hybrid version of a software agent with some intelligence of its own. “[An Intelligent Software agent is] a piece of software that performs a given task using information gleaned from its environment to act in a suitable manner so as to complete the task successfully. The software should be able to adapt itself based on changes occurring in its environment, so that a change in circumstances will still yield the intended result.” (Herman’s 1997)

2.3.2 Functions

Intelligent software agents should perform the following tasks continuously

1. Insight of changing environment
2. Action required for the change
3. Reason to the action taken
4. Solution for the problem
5. Draw Inferences and perform decision tree for future use.

2.4 Background and Related Work

Information sharing and Trust negotiation in intelligent agents have their root way behind from 90's. There are various researches going on Information sharing in Intelligent Software Agents lab of Carnegie Mellon (the Robotic Institute). One of such is Information sharing in Agents. They have alternative decision making systems and Bilateral Negotiations with outside options. In this chapter for knowing the background of trust negotiation, will discuss some of the points from the bilateral negotiation with outside options.

The bilateral negotiations paper considers each trust negotiation as a thread. The model is composed of three modules: single-threaded negotiations synchronized multi-threaded negotiations, and dynamic multi-threaded negotiations. The single-threaded negotiation model provides negotiation strategies without specifically considering outside options. The model of synchronized multi-threaded negotiations builds on the single-threaded negotiation model and considers the presence of concurrently existing outside options. The model of dynamic multi-threaded negotiations expands the synchronized multithreaded model by considering the uncertain outside options that may come dynamically in the future.

Most related work can be classified into one of several major categories. The first strand of research is based on a centralized model or distributed model where there are agents such as match maker, information broker or message broad who can response for all information communication [2, 3]. These works has been shown to be able to greatly improve multi-agent [4] system performance [5]. However, such work is inadequate for large team, since it is impossible or undesirable for all team members to share all their information all the time, i.e. because of the limit of required communication channels. The second major strand of research is relies on agents maintaining a shared model of each other or knowing exactly

other members' actual internal state as STEAM[6], COM-MTDP [7] and CAST [8]'s mental model. However, as for centralized approaches, in large team there is insufficient bandwidth to support such an approach.

The information sharing problem can also be handled by setting up decentralized model. Both [9] and [10] did a communication decision model based on Markov decision processes (MDP). Their basic idea is an explicit communication action will incur a cost and they supposed the global reward function of the agent team and the communication cost and reward are known. Moreover, [11] put forward a decentralized collaborative multi-agent communication model and mechanism design based on MDP which assumed that agents are full-synchronized when they start operating, but no specific optimal algorithm was presented. Unfortunately, there are no experimental results showing that their algorithm can work on large teams. Incomplete information theory is another way to solve the information sharing problems. [12] Presents a framework for team coordination under incomplete information based on the incomplete information game theory that agents can learn and share their estimates with each other. [13] Used a probability method to coordinate agent team without explicit communication by observing teammates' action and coordinating their activities via individual and group plan inference. Research on social networks began in physics [14, 15, 16], but since it has been applied in many areas though rarely in multi-agent work.

2.5 System Architecture

The system model for information sharing among large teams can perform distributed information sharing without the cost of maintaining accurate models of all the teammates. First, impose a network topology on the team members analogous to the social networks that exist in human societies. The key characteristic of this network model is that information exchange is based on peer to peer communication. Specifically limit agents to communicating directly with only a small percentage of the overall team.

Leveraging the team network, our basic approach like Figure1 is when an agent has a piece of information to communicate, it forwards that information to the direct acquaintance most likely to actually need that information or know who will. Then the acquaintance performs the same reasoning when it gets the information. After passing through hopefully, a small number of team members, information arrive at a team member that needs it.

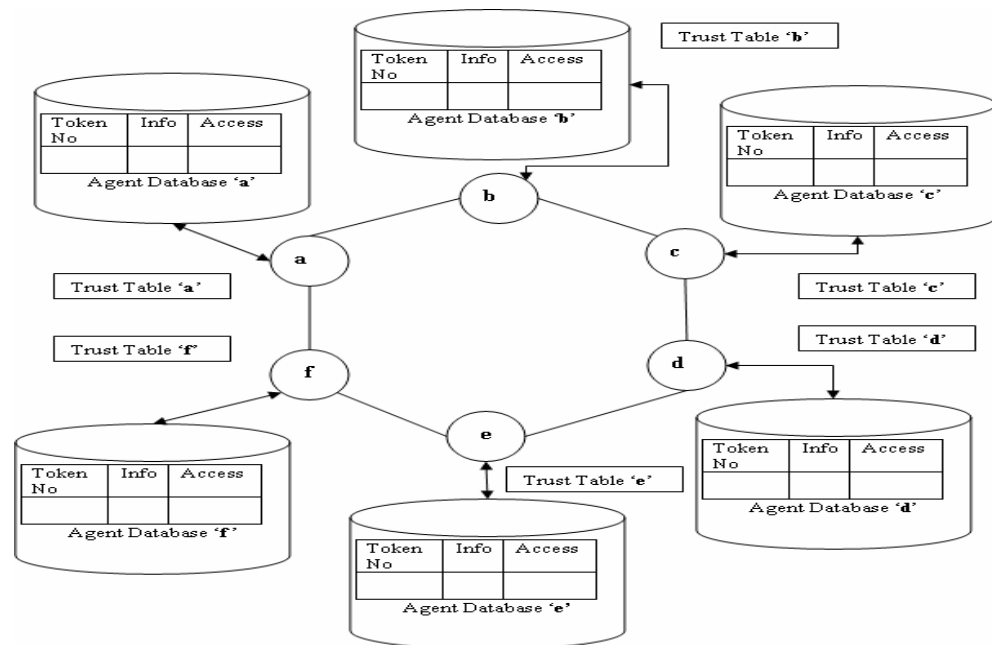


Figure 1. System Model for agent information sharing with confidentiality and trust negotiation

The intuition is that each agent attempts to guess which of its acquaintance either require the information or are in the best position to get the information to the agent that requires it. Even though members of large teams will not have accurate, up-to-date models of the team, our hypothesis is that the models will be accurate enough to deliver the information in a small number of “hops”. One agent is randomly chosen as the source of some information and another is randomly chosen as the sink for that information. A probability is attached to each link, indicating the chance that passing information down that link will get the information through the smallest number of links to the sink. The probability will increase as it reaches closer to the target. The chance of missing the target depends on the distance between the source and the sink. The number of “hops” to vary as the distance varies. The challenge is to construct complex models for information sharing but only have reasonable models to improve agent's guessing. The key question is how to create models that allow the agent to “guess” correctly more often than not. To achieve this, we observe that each piece of domain knowledge is typically related to each other piece of domain information. For example, if agent ‘a’ tells agent *b* about a plan to destroy an enemy base, when agent *b* gets the information that the base is fake, sending that information to agent *a* is a reasonable thing

to do, since *a* likely either needs the information or knows who does. So it is reasonable to infer from an agent's formerly sent message that it may need the other kind of information to improve the performance as the above example. Thus, the previously received information can be interpreted as evidence to infer which acquaintance to send other information to. If an agent maintains a knowledge base about what it heard from its acquaintances, it can use that knowledge to determine where to route newly received information. The other challenge in the network is trust management. Consider the previous example. In case if agent *a* is not trustworthy then that information to destroy the enemy base might be fake. So trust negotiation is an important goal. In our system we can negotiate trust based on their acquaintance. For instance the source is acquainted with another agent in the network that is acquainted with the sink; the sink can get the trust level from its acquaintance. In the beginning the complex network will be formed with no acquaintances. Then once the connection is setup and each agent begin to acknowledge each other's neighbors then the trust levels are assigned to the agent based on their information. If there is a bad agent then it tends to spoil the entire system. The other agent sends the bad acquaintance that they have had with the corresponding agent.

In the system simulation there is also a security policy implementation that has a very important part in the sharing of the information to authorized agents rather than transferring the D4 level data to lower access agents. The token and the information are linked with a security level. Each agent maintains its own level of confidentiality for any particular information. There may be instances where the same information with different clearance domain can be stored in different agents. This also makes a possibility that if one agent rejects the request based on the trust level of the requesting agent, another agent can service the request based on the trust level or acquaintance level that it has maintained with for that corresponding agent. The following example can explain the point. Agent '*a*' can have two or more acquaintances in this case it is two '*b*' and '*c*'. The trust level of '*a*' with '*b*' is in higher clearance domain say D3 and with '*c*' it is in D2. If there is a request from '*a*' sent for some information at D3 then '*c*' will reject the request and '*b*' will service the request. Similarly '*b*' and '*c*' have two different levels for the same information *i.e.* information '*x*' at D3 in '*c*' and at D2 in '*b*'. If '*a*' request for the same information then there is a chance that '*b*' will service the request.

2.6 Implementation of the System

2.6.1 Overview

The simulation of the intelligent agents sharing information is done using Java programming. The program mainly concentrates on two things. How much message is being transferred from each agent and the trust element within each agent? The summary of the simulation mainly has results on how much message each agent had in the beginning of the session? How much they shared with the other agents in the simulation and how much they received from the simulation. The important feature of the simulation is that it also holds the history of the summary which makes easy to know the amount of data lost in each session. The agents can make use of the history of the summary to learn more about the other agents in the simulation and try to avoid the more data loss in the future session with the same set of agents. This also helps in knowing the nature of the agents involved, if they are ready to participate and send more messages or they are just waiting to get the most out of the other agents. Such agents are also blocked from the simulation by not sending messages to that particular link. This depends on the individual discretion of the agents. They also pass on the information to other agents in the simulation that such a neighbor is not willing to send any information and readily accepts all the information that is passed on to it or through it. Those dormant agents are like leeches that spoil the entire network.

The agents in the simulation share the information upon request from any other agent in the network. The information with all the agents is inter-related. The messages are numbered in order so as to know the entire flow of the information. Each agent starts collecting the information from other agents based on the information that it has in hand, for example if an agent has a message and its part number 5. It doesn't have any other message numbered prior to 5 or after 5. So the agent first requests for 6 and 4. If it acquires 6 it sends out 7, if it gets 4 the next request is for 3. The agents continue the above way of request until they get whatever they needed from the entire information. The algorithm is explained clearly in the next section of Implementation.

2.6.2 Algorithm for Information sharing with Confidentiality and Trustworthy Computing

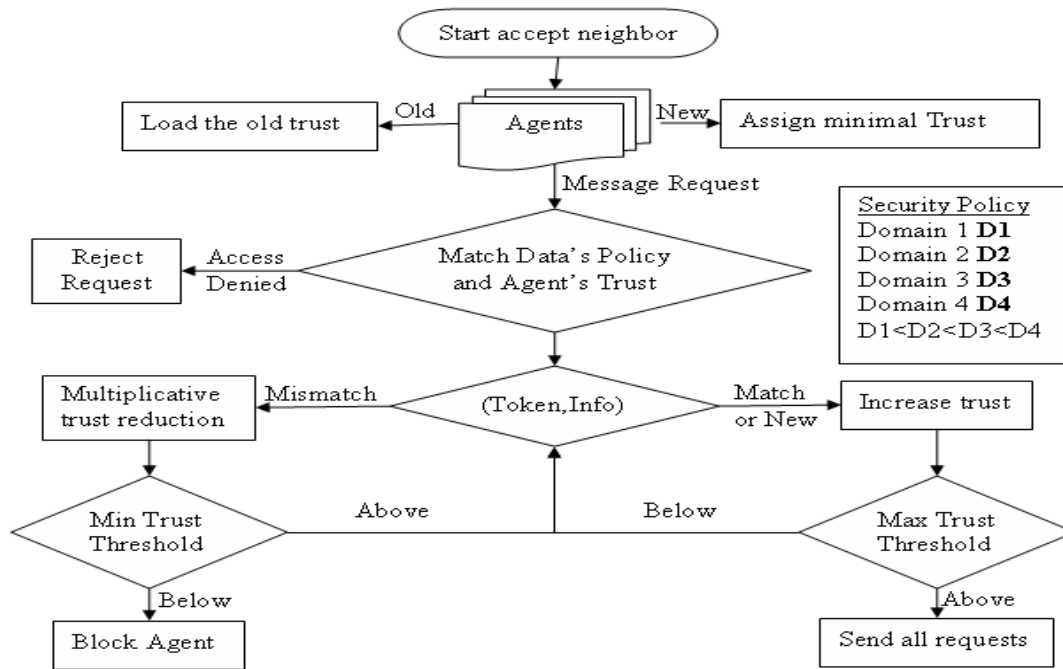


Figure 2. Flow diagram of the Information sharing with confidentiality and trust management.

In this algorithm as in Figure 2, at the time of forming the coalition, the agents have the information about the direct acquaintances i.e. a neighbor and their trust level. If there is going to be a new neighbor the trust level is set to a minimum acquaintance level. Then each agent has its own set of information to be shared with other agents in the network. The information is linked with a significant token number and a security Policy. The moment a message is requested by some agent for some information, the token is received then the security policy of the corresponding information is matched with the clearance domain of the requesting agent. The clearance domain mainly depends on the trust level of the requesting agent that is linked with the source agent. In this algorithm there are four such clearance domains: D1, D2, D3, D4 and we assume for simplicity $D1 < D2 < D3 < D4$. The trust levels are similarly split into four levels where in the minimum threshold is set for D1 information (that is information in domain D1). We assume that each agent can read information at all

domains, however the trust level that one agent has on the other will determine the domain information that an agent sends to another agent.

There may be multiple copies of the information existing simultaneously in the network along with the same token number, yet the token and information pair is always unique. If the agent gets the same information with two different tokens or vice versa, then his discrepancy will lead to loss of trust. It will perform a multiplicative decrease in the trust level. Similarly if new information arrives trust level of the acquaintance is increased. There is a minimum and maximum threshold level for trust. If any acquaintance falls below the minimum threshold of the trust, then they are removed from the circle of trust, further communication is stopped and the rest of the acquaintances are notified about the bad agent. If the acquaintance's trust level goes above the maximum threshold then the agent sends all the messages requested by the acquaintance. The information sharing goes on until one agent gets the entire information it needs or to a fixed number of time where all the agents have the list of data lost and data gained. It also stores the history of the direct acquaintance and its trust level which helps in future coalition with the same agent.

2.6.3 Specifications of Algorithm

- Form Communication link with other agents where the neighbors are the acquaintances.
- If new neighbor set minimal trust level else load the existing trust level from the database.
- If an agent request for some information. Check the trust level of the agent and the access or security level linked with the information
- If the access is granted allow service the request based on priority. Else reject request.
- Start sending and receiving messages (the tokens and the Information are linked).
- If there is mismatch in messages multiplicative decrease of trust and if the trust goes below minimal trust after decreasing block agent and notify the network
- If there is message (new or old with match) additive increase trust and also if the trust is above max threshold send the entire request one by one.
- If any one agent has all information or end of session occurs end link store trust level, Message (Token and Information).
- Calculate the amount of data lost or gained from each acquaintance

2.7 Experimental Results

The simulation of the algorithm was implemented and there were many sets of results generated. The experimental results were very much helpful in understanding how the system works. In the below chart 1 the Information that was sent from each agent and the information gathered at each end is collected and the Net Gain is also calculated.

Let $T \rightarrow$ Net Gain/Loss of Information for any agent.

$R \rightarrow$ The message received from Agents by some agent a_i .

$S \rightarrow$ The message sent to other Agents ($a_0, a_1 \dots \dots a_n$) by agent a_i .

$O \rightarrow$ The own message of each agent in the beginning of the session.

$$T = (R - (S + O))$$

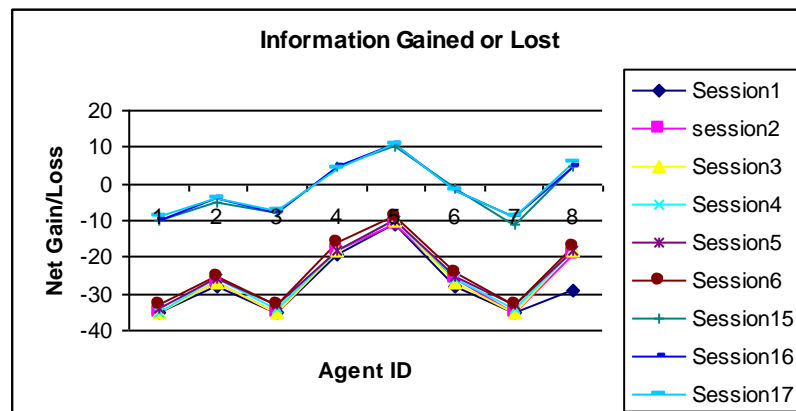


Chart 1. Net gain of information by each agent in four continuous sessions of information sharing with trust computing.

The Chart 1 has a set of simulations that was done within 8 agents. The simulation revealed that as the session increases the gain also increases. This is because the agents come to know well about the other agents. Agents have the trust level of each agent in their database summary. The trust level increases the gain increases. Since most of the agents send 90% non negative messages the gain increases for each session. Gain is the amount information collected from other agents apart from the ones it has forwarded to other agents without storing it in its database, for example if an agent has 10 messages with it before the start of the session, it receives 25 messages from other agents of which its request is 15 messages.

The number of messages sent by the corresponding agent to other agents in the session in

response to their request is considered as loss of information say 5. The net gain is $(25 - (15 + 5))$ is equal to 5. The chart1 also clearly shows that there is not a much of difference in gain with each agent in successive session. They all share same level of trust in the beginning and the gain varies based on their trust level through the simulation. If they send one negative message their gain goes down. The neighbors stop sending messages if they are notified that some agent is below the threshold level of some other acquaintance. So the gain in sharing depends mainly on the trust level. The trust of the node is directly proportional to the quality of the message sent by that agent and the gain is also directly proportional to the trust achieved by the agents. The chart has net gain and loss on its Y axis. The series one to four indicate the simulation that was conducted on the eight agents in continuous session of information sharing. The chart clearly indicates the increase in gain as the session progresses. The level of gain varies from one agent to another because of the neighbors, amount of information the neighbors possess with them. For instance in the chart agent 5 has its gain level in -10 and rest of the agents have it in -25 to -35. The reason is due to the neighbors of the agents are 4 and 8. So the information provided by the neighbors gives more gain to agent 5 than other agents. 4 and 8 also in turn get more gain from 5 and stand higher than the other agents. The summary of the experimental results contain the amount of message sent, received and the Net. It also has the recent trust level of all the neighbors. The newly received tokens are also copied in the summary.

2.8 Summary and Directions

The proposed algorithm has been implemented. The experimental results show that the information sharing is done as in peer to peer communication network. The amount of information lost and gained is stored at the end in the database. The number of messages sent to share a little amount of information through the network is high. The scalability also becomes an issue. If there are more neighbors the amount of message sent and managing the traffic of messages becomes a very big issue. The future work on this research can be implementation of the above system in which the guess and hops are calculated to the efficient way to share information among the agents.

A major issue we leave for future research is how to calculate the relationships between pieces of information which is highly relative with domain knowledge and expertise where

our algorithm should be applied. Furthermore, we do not investigate how information sharing works on negative relative messages where the relationship between pieces of information. Does the dormant agent gain more than the other active agents? Can the agents form a multicasting group which might help in communicating with a group of agents simultaneously? The multicasting group will save a lot of network resources by sending one message to a gateway agent and thereby pass it to the whole multicast group.

REFERENCES

- [1] P. Scerri, Y. Xu, E. Liao, J. Lai, M. Lewis, K. Sycara. Coordinating very large groups of wide area search munitions, Recent Developments in Cooperative Control and Optimization, Dordrecht, NL: Kluwer Academic Publishers.
- [2] M. H. Burstein and D. E. Diller. A framework for dynamic information flow in mixed-initiative human/agent organizations. Applied Intelligence on Agents and Process Management, 2004. Forthcoming.
- [3] K. Decker, K. Sycara, A. Pannu and M. Williamson. Designing behaviors for information agents. Procs. Of the First International Conference on Autonomous Agents, Feb., 1997.
- [4] P. R. Cohen, H. J. Levesque and I. Smith. On team formation. In J. Hintikka and R. Tuomela, editors, Contemporary Action Theory, Synthese, 1998.
- [5] K. C. Jim and C.L. Giles. How communication can improve the performance of multi-agent systems. In Proceedings of Autonomous agents'01, 584-591, 2001.
- [6] P. Scerri, Y. Xu, E. Liao, J. Lai, K. Sycara. Scaling Teamwork to Very Large Teams, AAMAS 04, Forthcoming, 2004.
- [7] D. Pynadath and M. Tambe. The communicative multiagent team decision problem: analyzing teamwork theories and models. Journal of Artificial Intelligence Research, Vol.16, pages 389-423, 2002.
- [8] J. Yen, J. Yin, T. R. Ioerger, M. S. Miller, D. Xu and R. A. Volz. Cast: Collaborative agents for simulating teamwork. In Proceedings of IJCAI'01, pages 1135-1142, 2001.
- [9] P. Xuan, V. Lesser and S. Zilberstein. Communication decisions in multiagent cooperation: Model and experiments. In Proceedings of Autonomous Agents'01, 2001.

[10] C.V. Goldman and S. Zilberstein. Optimizing information exchange in cooperative multi-agent systems. Proceedings of the Second International Conference on Autonomous Agents and Multi-agent Systems, 2003.

[11] C.V. Goldman and S. Zilberstein. Mechanism design for communication in cooperative systems. Game Theoretic and Decision Theoretic Agents Workshop at AAMAS' 03, July, 2003.

[12] H.H. Bui, S. Venkatesh and D. Kieronska. A framework for coordination and learning among team members. In Proceedings of the Third Australian Workshop on Distributed AI (DAI-97), pages 116-126, Perth, Australia.

[13] M.V. Wie. A probabilistic method for team plan formation without communication. Proceedings of the Fourth International Conference on Autonomous Agents, pages 112-113, Barcelona, Spain, June 3-7, 2000.

[14] R. Albert and A. Barabasi. Statistical mechanics of complex networks. Review Modern Physics, 74, 47,2002.

[15] M. E. J. Newman. The structure and function of complex networks. SIAM Review, Vol. 45, No. 2, pages 167-256, 2003.

[16] D. Watts and S. Strogatz. Collective dynamics of small world networks. Nature, 393:440-442, 1998.

[17] B. Thuraisingham, Assured Information Sharing, UTD Technical Report, December 2006 (also to appear as Book Chapter in Data Mining for Security Applications edited by H. Chen et al)

[18]. B. Thuraisingham, Database and Applications Security, Integrating Data Management and Information Security, CRC Press, 2005.

CHAPTER 3
EXTENDED RBAC –BASED DESIGN AND IMPLEMENTATION FOR A SECURE
DATAWAREHOUSE

Srinivasan Iyer and Dr.Bhavani Thuraisingham

The Department of Computer Science

The University of Texas at Dallas

P.O. Box 830688

Richardson, Texas 75083-0688

Copyright© 2007 IEEE. Reprinted with permission from Second International Conference
on Availability, Reliability and Security. ARES 2007

3.1 Abstract

This chapter first discusses security issues for data warehousing. In particular, issues on building a secure data warehouse, secure data warehousing technologies as well as design issues are discussed. Our design of a secure data warehouse that enforces an Extended RBAC Policy is described next. Finally directions for secure data warehouses are discussed.

3.2 Overview

Data warehousing is one of the key data management technologies to support data mining and other decision support functions. Several organizations are building their own warehouses. Commercial database system vendors are marketing warehousing products. As stated in [INMO93], the idea behind a data warehouse is that it is often cumbersome to access data from the heterogeneous databases. Several processing modules need to cooperate with each other to process a query in a heterogeneous environment. Therefore, a data warehouse will bring together the essential data from the heterogeneous databases. This way the users need to query only the warehouse. Essentially data warehouses provide support for decision support of an enterprise. For example, while the data sources may have the raw data, the data warehouse may have correlated data, summary reports, and aggregate functions applied to the raw data.

Now, in order for the data warehouse to be useful in many applications such as medical, financial, defense and intelligence, it must be secure. In other words the data warehouse must enforce the security policies enforced by the back-end data sources in addition to possibly enforcing additional security properties. Figure 3 illustrates a high level view of a secure data warehouse. The data sources are managed by secure database systems A, B, and C. The information in these secure databases are merged and put into a secure warehouse.

In this chapter we discuss security for data warehousing. The organization of this chapter is as follows. Some issues on building a secure warehouse is discussed in Section 2.3 which also includes the design issues for a secure data warehouse. Next our design and implementation of a secure warehouse based on an extended RBAC model will be discussed in section 2.4. Directions are discussed in section 2.5.

3.3 Security for Data Warehouses

3.3.1 Security Issues

There are various ways to building a secure data warehouse. One is to simply replicate the secure databases and enforce an integrated security policy. This does not have any significant advantage over accessing the secure heterogeneous databases. The second approach is to replicate the information, but to remove any inconsistencies and redundancies. This has some advantage, as it is important to provide a consistent picture of the databases. The third approach is to select a subset of the information from the databases and place it in the warehouse and at the same time ensuring that security is maintained by the warehouse. There are several issues here. How are the subsets selected? Are they selected at random or is some method used to select the data? For example, one could take every other row in a relation (assuming it is a relational database) and store these rows in the warehouse. The fourth approach, which is a slight variation of the third approach, is to determine the types of queries that users would pose, and then analyze the data, examine security policies to be enforced and store only the data that is required by the user. We will call this secure on-line analytical processing (SOLAP) as opposed to secure on-line transaction processing (SOLTP) where the back-end secure database systems are queried.

With a data warehouse, data may often be viewed differently by different applications. That is, the data is multidimensional. For example, the payroll department may want data to be in a certain format while the project department may want data to be in a different format. The warehouse must provide support for such multidimensional data. Furthermore different security policies may be enforced at different levels. For example, only managers can see the individual salaries while the project leaders see average salaries.

In integrating the data sources to form the warehouse, a challenge is to analyze the application and select appropriate data to be placed in the warehouse. At times, some computations may have to be performed so that only summaries and averages are stored in the data warehouse. Note that it is not always the case that the warehouse has all the information for a query. In this case, the warehouse may have to get the data from the heterogeneous data sources to complete the execution of the query. Another challenge is what happens to the warehouse when the individual databases are updated? How are the updates propagated to the warehouse? How can security be maintained when propagating the

updates? These are some of the issues that are being investigated. Security cuts across all layers and operations of the warehouse. In [THUR97] we discussed security for data warehousing. Since then there have been some efforts on secure data warehouses (see for example [BEST02]).

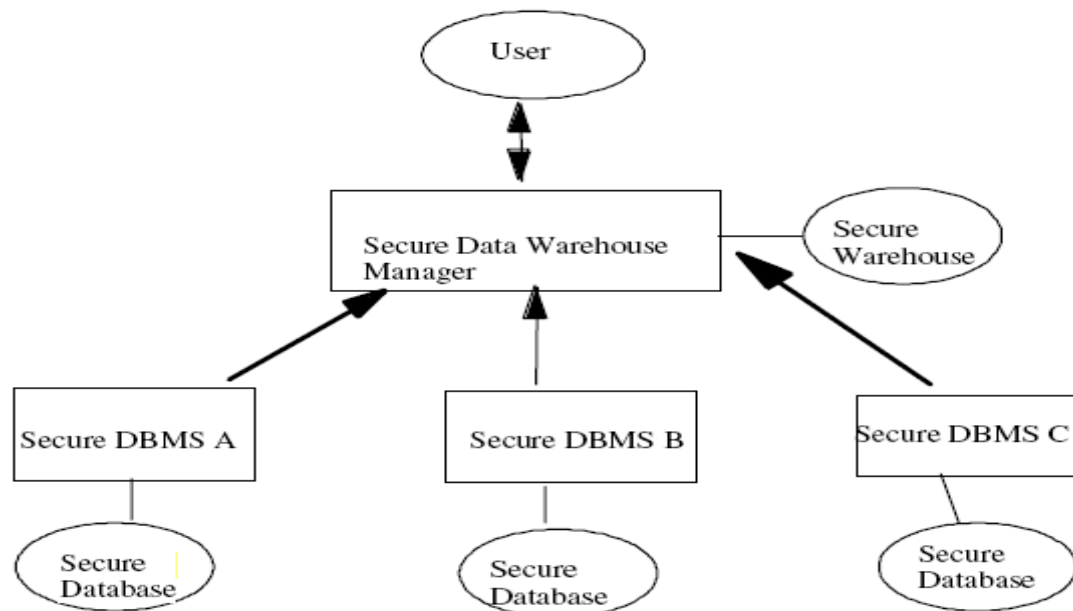


Figure 3. Secure Data Warehouse Example

One of the major security and privacy challenges for data warehousing is the inference and privacy problem. For example, the data warehouse aggregates the data. Therefore the aggregated data could be highly sensitive or private while the individual data values may be unclassified or public. Various privacy preserving data mining and data warehousing approaches are being investigated [THUR05], [THUR03].

3.3.2 Technologies for Secure Warehousing

Note that several secure information technologies have to be integrated to develop a secure data warehouse. These include secure heterogeneous database integration, statistical databases, secure data modeling, secure metadata management, secure access methods and indexing, secure query processing, secure database administration, general database security, and secure high performance database management.

Secure heterogeneous database integration is an essential component to data warehousing. This is because data from multiple secure heterogeneous data stores may have to be integrated to build the warehouse. In the case of secure heterogeneous database integration discussed in [THUR94], there is usually no single repository to store the data. However, in a secure warehouse there is usually a single repository for the warehouse data and this repository has to be managed and security policies enforced.

Statistical databases keep information such as sums, averages, and other aggregates. There are various issues for statistical databases. For example, how can summary data maintained when the database gets updated? How can the individual data items be protected? For example, the average salary may be Unclassified while the individual salaries are Secret. Since warehouses keep summary information, techniques used to manage statistical databases need to be examined for warehouses.

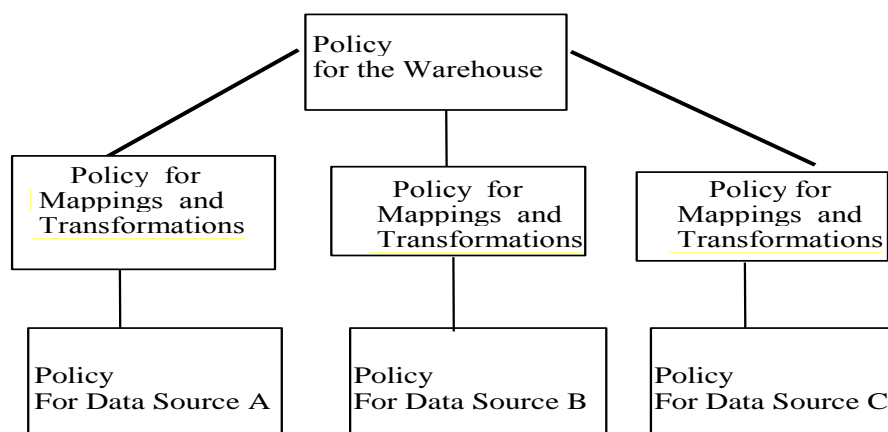


Figure 4. Security Policy Integration

Secure data modeling is an essential task for building a data warehouse. Is the secure data model influenced by the data models used by the back-end secure heterogeneous data sources? Should a data model be developed from scratch? Inmon has outlined several steps to developing a data model [INMO93]. He says that at the higher level there are three stages: developing a corporate model, an enterprise model, and a warehouse model. At the middle level there may be a model possibly for each subject, and at the physical level it includes features such as keys. Some argue this is too lengthy a process and that one should get to the warehouse model directly. As more experiences are reported on developing data warehouses,

this issue may be resolved. New types of data models such as multidimensional data models and schemas such as star-schemas have been proposed for data warehousing. We need to integrate these models with secure data models that we have discussed in [THUR93a]. For example, in a project database, there is a central table that has key information on projects such as project number, project leader, estimated time duration, cost and other pertinent data. Each of the entries in this table could be elaborated in other tables. For example, estimated time duration could be in days, months, and years. Cost could be dollars, pounds, yens and other currency. Depending on who is using the data, different views of the data could be provided to the user.

Appropriate access methods and index strategies have to be developed for the warehouse. For example, the warehouse is structured in such a way so as to facilitate query processing. An example query may be: how many red cars costing more than 50K were bought in 1995 by physicians? Many relations have to be joined to process this query. Instead of joining the actual data, one could get the result by combining the bit maps for the associated data. The warehouse may utilize an index strategy called a bit map index where essentially there is a 1 in the bit map if the answer is positive in the database. So, if the color of the car is red, then in the associated bit map, there will be a 1. This is a simple example. Current research is focusing on developing more complex access methods and index strategies. We need to examine the security impact on query processing strategies for the warehouse. For example, does query modification apply for secure warehousing? Suppose the user is not able to see the sales figures for those living in region X. Then the query has to be modified as follows: How many red cars costing more than 50K did physicians who do not live in region X buy in Detroit in 1995?

Developing an appropriate query language for the warehouse is an issue. This would depend on the data model utilized. If the model is relational, then an SQL-based language may be appropriate. We then need to examine extending SQL to specify security constraints such as User group A cannot see any information about the purchase of red cars by physicians from region X. One may also need to provide visual interfaces for the warehouse.

Secure database administration techniques may be utilized for administering the warehouse. Is there a warehouse administrator? What is the relationship between the warehouse administrator and the administrator of the data sources? How often should the

warehouse be audited? Another administration issue is propagating updates to the database. In many cases, the administrators of the data sources may not want to enforce triggers on their data. If this is the case, it may be difficult to automatically propagate the updates. What is the security impact on update propagation? What are the functions of the Systems Security Officer (SSO) for the warehouse? Should there be a Warehouse Security Offer (WSO)?

Security solutions for integrating heterogeneous and federated database systems discussed in [THUR94] may be applied to secure data warehouses. For example, we need to examine the challenges for secure federated database management to integrate the security policies for data warehousing. Figure 4 illustrates security policy integration for data warehousing. We need to develop secure transformations as we move from one layer to the next in building a warehouse.

As stated earlier, statistical database security is one of the technologies for securing the data warehouse. Since the warehouse gives out sums and averages, how can one protect the sensitive values from which the sums and averages are computed? Security controls also have to be enforced in maintaining the warehouse as well. This will have an impact on querying, managing the metadata, and updating the warehouse. In addition, if multilevel security is needed, then there are additional considerations. For example, what are the trusted components of the warehouse?

High performance computing including parallel database management plays a major role in data warehousing. The goal is for users to get answers to complex queries rapidly. Therefore, parallel query processing strategies are becoming popular for warehouses. Appropriate hardware and software are needed for efficient query processing. We need to integrate security into parallel database systems. Some preliminary work was reported in [THUR93b]. We need to carry out further investigations.

Secure metadata management is another critical technology for data warehousing. The problem is defining the metadata. Metadata could come from the data sources. Metadata will include the mappings between the data sources and the warehouse. There is also metadata specific to the warehouse. We need to examine the security impact on metadata management. There are three types of metadata. One is metadata for the individual data sources. The second is the metadata needed for mappings and transformations to build the warehouse, and the third is the metadata to maintain and operate the warehouse.

Secure distributed database technology discussed in [THUR91] plays a role in data warehousing. Should the warehouse be centralized or distributed? If it is distributed, then much of the technology for secure distributed database management discussed [THUR91] is applicable for data warehousing. In the non distributed case, there is a central warehouse for the multiple branches, say in a bank. In the distributed warehouse case, one may assume that each bank has its local warehouse and the warehouses communicate with each other.

3.3.3 Design Steps

Designing and developing the secure data warehouse is a complex process and in many ways depends on the application. A good reference to data warehousing is the book by Inmon [INMO93]. It describes the details of the issues involved in building a data warehouse. In this section we outline some of the steps to designing the secure warehouse. Figure 5 illustrates some of these steps. There are three phases to developing a secure warehouse. One phase focuses on structuring the secure warehouse so that secure query processing is facilitated. In other words, this phase focuses on getting the data out of the warehouse. Another phase focuses on bringing the data into the warehouse. For example, how can the secure heterogeneous data sources be integrated so that the data can be brought into the warehouse and yet security be maintained? The third phase maintains the warehouse once it is developed. This means the process does not end when the secure warehouse is developed. It has to be continually maintained. We first outline the steps in each of the phases.

One of the key steps in getting the data out of the warehouse is application analysis. For example, what types of queries will the users pose? How often are the queries posed? Will the responses be straightforward? Will the users need information like summary reports? A list consisting of such questions needs to be formulated. Furthermore, we need to examine the security constraints enforced by the warehouse and determine how these constraints may be enforced.

Another step is to determine what the user would expect from the warehouse. Would he want to deal with a multilevel relational model or a multilevel object-oriented model or both? Are multiple views needed? How can access be controlled to the views? Once this is determined, how do you go about developing a secure data model? Are there intermediate models?

A third step is to determine the metadata, index strategies, and access methods. Once the query patterns and data models have been determined, one needs to determine what kinds of metadata have to be maintained. What are the index strategies and access methods enforced? What are the security controls on the index strategies and access methods?

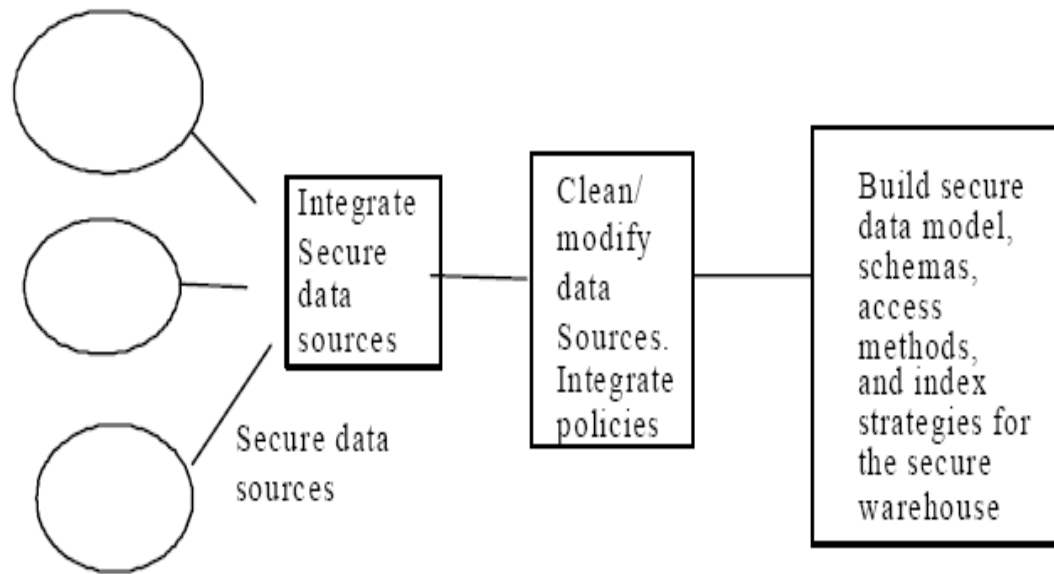


Figure 5. Developing a secure data Warehouse

A closely related task is developing the various schemas and policies for the warehouse. Note that the individual databases will have their own schema and security policies. The complexity here is in integrating these schemas to develop a global schema for the warehouse. While schema integration techniques for distributed and heterogeneous databases may be used, the warehouse is developed mainly to answer specific queries for various applications. Therefore, special types of schemas such as star schemas and constellation schemas have been proposed in the literature. Products based on these schemas have also been developed. However we need to take a closer look at the schemas and examine the security impact. Furthermore we need to explore techniques to integrate the security policies.

There are several technical issues in bringing the data into the warehouse from the different data sources. What information should be deleted from the individual databases when the data is migrated to the warehouse? How should integrity be maintained? What is the security

policy? How can inconsistencies be resolved? For example, we need to ensure that by querying the warehouse the sensitive information in the back-end databases is not revealed to the user who does not have proper access control to this information. This requires a lot of work. Various algorithms for integrating heterogeneous databases have to be examined. At the end of this stage, one would have some form of a secure warehouse. Multi-tier architecture is becoming popular for data warehousing. Essentially, data passes through multiple tiers before reaching the warehouse. We need to examine the security impact on multi-tier architecture. At the bottom tier are the data sources. At the top tier is the data warehouse. Between the top and bottom there may be multiple tiers. Each tier has its own schemas, metadata, and various administration details as well as policies, and each tier takes advantage of the work done at lower tiers.

Once the secure warehouse is designed and developed, there are also some additional considerations for maintaining the warehouse. How is the security of the warehouse maintained? Should the warehouse be audited? How often is the warehouse updated? How are the changes to the local databases to be propagated to the warehouse? What happens if a user's query cannot be answered by the warehouse? Should the warehouse go to the individual databases to get the data if needed? How can data quality and integrity be maintained?

We have outlined a number of phases and steps to developing a secure data warehouse. The question is, should these phases and steps be carried out one after the other or should they be done in parallel? As in most software systems, there is a planning phase, a development phase, and a maintenance phase. However, there are some additional complexities. The databases themselves may be migrating to new architectures or data models. This would have some impact on the warehouse. New databases may be added to the heterogeneous environment. The additional information should be migrated to the warehouse without causing inconsistencies. These are difficult problems and there are investigations on how to resolve them. Although there is much promise, there is a long way to go before viable commercial secure data warehouse products are developed.

In summary, a secure data warehouse enables different applications to view the data differently and at the same time enforce the security policies. That is, it supports multidimensional and multilevel data. Data warehouse technology is an integration of

multiple technologies including heterogeneous database integration, statistical databases, and parallel processing. The challenges in data warehousing include developing appropriate data models, architectures (e.g., centralized or distributed), query languages, and access methods/index strategies, as well as developing techniques for query processing, metadata management, maintaining integrity and security, and integrating heterogeneous data sources. Integrating structured and unstructured databases, such as relational and multimedia databases, is also a challenge. Security cuts across all layers and all stages of the development.

While the notion of data warehousing has been around for a while, it is only recently that we are seeing the emergence of commercial products. This is because many of the related technologies such as parallel processing, heterogeneous database integration, statistical databases, and data modeling have evolved a great deal and some of them are fairly mature technologies. Furthermore secure database technology is also fairly mature. There are now viable technologies to build a secure data warehouse. We expect the demand for secure data warehousing to grow rapidly over the next few years.

It should be noted that many of the developments in data warehousing focus on integrating the data stored in structured databases such as relational databases. In the future we can expect to see secure multimedia data sources being integrated to form a warehouse.

3.4 Design of Extended RBAC for Secure Data Warehousing

3.4.1 Overview

Data warehousing needs end to end security because, the entire data warehousing environment is not just the database, it as an enterprise wide system where there is an operation system from which data is extracted, a transformation system which transfers the data to the data warehouse and possible distribution to various other data marts or end users. Data management and data mining is an integral part of any corporation. The efficient management of the data allows increasing the performance of the entire corporation. On the whole data warehousing is a very complex part, it has multiple data sources, numerous applications and various end-users. It has to be prevented from being hacked by illegitimate users, preventing the access of data by unintended users and protect the data from damages and modification by them.

There can be various organization involved in corporation, each organization should have a control over their data and should be able to release whatever they like to share it in the corporation. In the previous section we discussed in detail about the existing security components for data warehouses. Now in this section we will look into RBAC in Data warehousing, the issues in the existing model, new system design, advantages over the existing model, Implementation issues.

In this section we discuss the design and implementation of an extended RBAC model for secure data warehouse. First in 2.4.2 section we provide an overview of RBAC and describe its limitations, then in section 2.4.3 we discuss the Extended RBAC model as well as the design and implementation of a secure data warehouse, experiments and challenges.

3.4.2 Role Based Access Control for Data Warehousing

3.4.2.1 Overview

The corporations have strict laws to maintain privacy and confidentiality. Every organization is willing to spend enormous amount of money in identity management which is also an integral part of data warehousing. There are can be scenarios where the private data can be in the database that is being mined by different organizations in the corporation. The security model should be easy to implement, low in maintenance, should administrate access controls ensure network and data security. While RBAC can be challenging to design and implement, it can be tailored to a company's business model and security risk tolerance. Once implemented, it scales for growth and requires minimal maintenance.

Once all of the employee roles are populated into the database, role-based rules are formulated and workflow engine modules are implemented. Through these elements, role-based privileges can be entered and updated quickly across multiple systems, platforms, applications and geographic locations RBAC provides company wide control process for managing data and resources. RBAC systems also can be designed to maximize operational performance, maintain data consistency and integrity. They can streamline and automate many transactions and business processes and provide users with the resources to perform their jobs better, faster and with greater personal responsibility. With an RBAC system in place, organizations are better positioned to meet their own statutory and regulatory requirements for privacy and confidentiality, which is crucial for data management, as well

as requirements imposed by external business partners and government agencies. Directors, managers and IT staffers are better able to monitor how data is being used and accessed, for the purpose of preparing more accurate planning and budget models based on real needs.

3.4.2.2 Issues with the Existing Models

In an Enterprise wide data warehousing system the security component that has been widely in use is the role based access control. It is a traditional access control model, with strong administrative security. Role based access control is capable of handling an entire system, but there are certain issues that have been identified in RBAC. The definition of users and groups are not clear. Duties are not defined along with the roles. The RBAC model mainly arises based on the roles that naturally exist in the system. It has certain limitation with respect to maintaining temporal dependencies. Incase if there is need for an order of causality in some process the RBAC model will have a difficulty to find out which process has to occur first and which should continue or which one should be revoked. The decision process is very weak in RBAC. It does not take into consideration the pre-approvals needed for any process, it also does not have a component to decide the on-going approvals. The decision for any process is decided before the start of the process, which limits the system performance. Incase some process needs more privilege during the process or some object has to change its attribute say one user trying to extract multiple data source simultaneously and needs authorization of a data which was not approved at the beginning of the session, then the process has to be terminated and restarted again. It does not have mutable attributes. There are certain objects which need to change its attribute definition to support the smooth running of the system; such mutable attributes are not available in RBAC. To summarize the issues in using RBAC in Data warehousing are User/Group definition, Temporal Dependencies, Mutable attributes and Decision process.

3.4.3 ERBAC System Architecture for Data Warehouses

3.4.3.1 Overview of ERBAC

The existing systems mostly use RBAC which has more limitation with respect to resource management. It has many issues regarding decision process, multiple roles, multiple session and many other temporal dependencies. So the warehouse needs are not fully met only

having RBAC has a security model. UCON is one of the modern security models which covers most of the traditional access model functionalities and has more new functions. UCON cannot exist alone and manage data security. It is not one for all complete solution. UCON has to co exist with some other traditional component in order to provide a strong secured data warehouse because it's a specific component which is mainly strong only in decision factor. The delegations of role are derived from traditional access control lists. The administrative functions are not as robust as in RBAC. So RBAC and UCON combination will form a strong access and usage control security component. This newly proposed component is E-RBAC (Extended Role Based Access Control). For details of RBAC and UCON we refer to [PARK04], [SAND96].

3.4.3.2 System Architecture of ERBAC

The system architecture of the Extended RBAC is similar to that of the existing security model. It is going to be a combination of the RBAC security component and UCON security model. The Administrative Security, Role delegations are part of RBAC component and the rights of objects and decision process is a combination of UCON. It will take into account the obligation approvals for pre and on-going transactions. It will also check the environment conditions before it gives the approval for authorization and the predicates of the authorization the obligations. The UCON model also helps in tracking the temporal dependencies there by helping to know if the current system can grant or revoke the operation that is going to be performed or that is already being performed. This component is not available in the existing role based access control. Combining both the models we get a secured system which encompasses both a highly secured administration rules. It also helps in the protection of identity management. The rights manager helps in checking the privacy constraints of the corresponding data.

In the architecture shown in figure 6 we see the architecture of the newly proposed system RBAC with UCON. The Administrator component provides the extension of role based access and usage control. The architecture also shows the imaginary division between the RBAC and UCON components. There are five managing components as shown in Figure 6: User Manager, Role Manager, Decision Manager, Session Manager and Data Manager. The administrator is the configuration controller which manages all the security components.

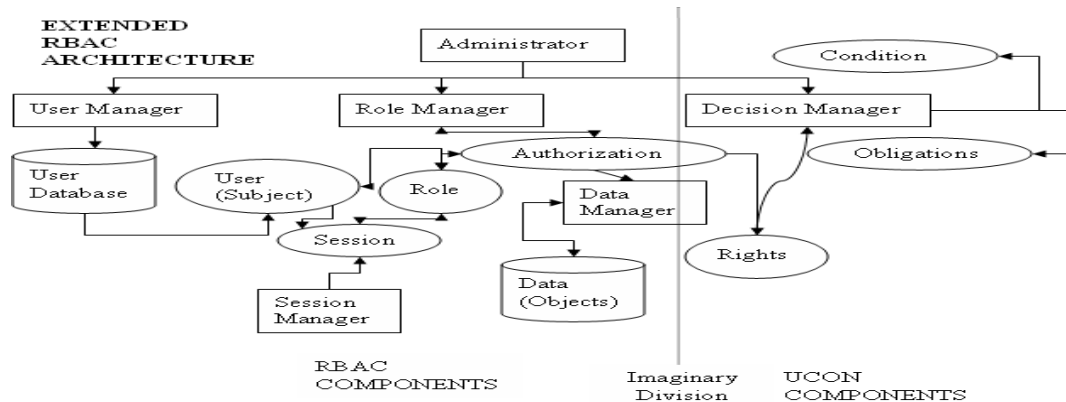


Figure 6. ERBAC Architecture with UCON_{ABC} Extension

User manager takes care of the list of users in the database using the system. It interacts with the role manager and gets the corresponding role of the user for the respective session. The session manager helps in maintaining multiple sessions and maps the history of the user in each session. Each time the user requests for some data the UCON comes into effect. The decision manager comes into the context and checks for the condition of the current system and check if it is going to be consistent even after the request being served. The Obligations are checked and the rights for the corresponding data are checked for pre approval and on-going approval. When the predicate approval is done depending on the role of the user the authorization is done. If all these decision process are checked then the user request is serviced. Thus the role based access and usage control is setup in a same system which has a strong administrative and decision process with temporal dependencies, mutability and identity management.

3.4.3.3 Implementation of ERBAC in Data Warehousing

Data warehousing is a complex system, it needs a proper security component that can ensure the safety of the entire system. The ERBAC model used in the data warehouse system mainly has an administrator component which manages the entire system security. The administrator component is linked from the beginning of a process through the end of the process. It clearly mentions the session details and also maintains history of each multiple session, multiple users in the same session and single user in the multiple sessions simultaneously.

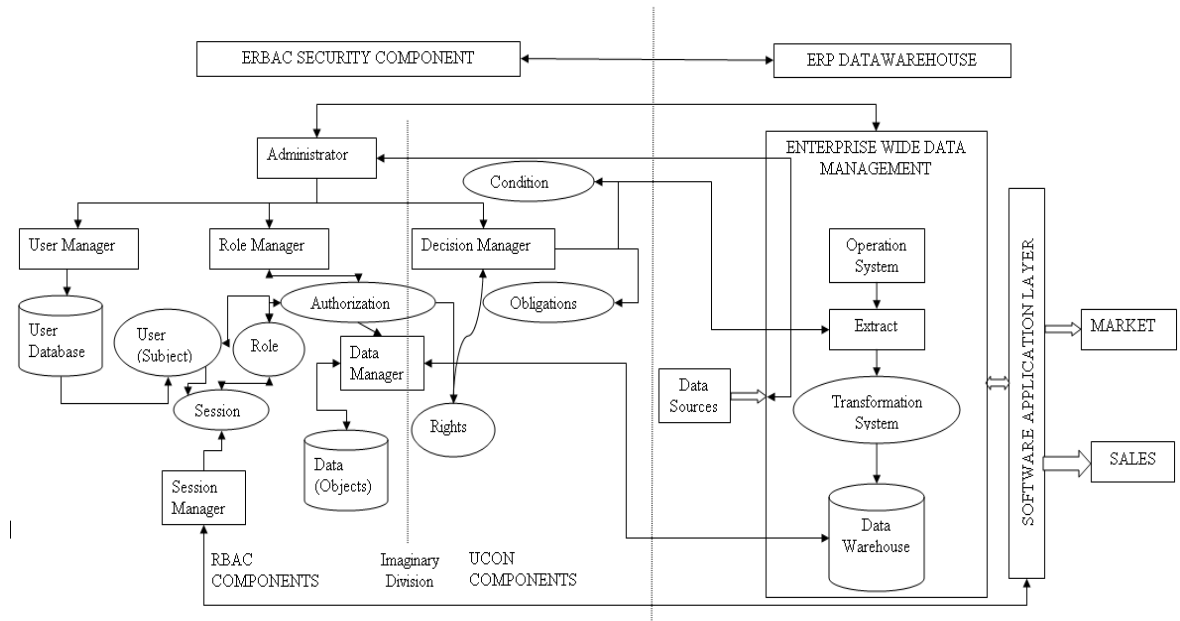


Figure 7. ERBAC in Enterprise Wide Data Warehouse

In a typical data warehouse shown in figure 7 the data is supplied by various data sources. The ERBAC component helps for each of the data sources to define the set of users/groups, their access privileges. The release attributes rights, coalition access policy. Once the data sources are authenticated, they are available for the enterprise through a secure connection. The data sources provide their rights, policy, access privileges to the administrator components which in turn send it to the usage control component. The usage control component has a rights manager and manages policies for the individual data sources and the corporation.

Once a user wants to make use of the data in the enterprise, he has to login to have a secure session and to provide him with the history of the previous sessions that he has been involved. The process is managed by the session manager component of ERBAC. It also allows the user to participate in the existing session and single user in multiple sessions. Once the session is created the process of accessing the enterprise system is initiated, the operations system access the data source which in turn access the rights manager gets the pre-approvals needed for the corresponding process. If the pre-approvals are granted the process loads the required data and the object attributes required for the process based on the access level of the user and the usage limit on the data for the corresponding user.

Once the pre-approvals are authenticated the process begins and while the process is in

progress there is a chance that some of the mutable attributes change the definition, in such cases the rights manager is accessed and the decision process also plays an important role, the rights manager checks if the corresponding attribute is allowed to change and if the change affects the system condition. If the system is consistent and the object is granted for the on-going approval, the process continues or if the grant is revoked then the process terminates.

Once the operations system gets the access it mines the data sources checks for the required pattern extracts and sends the data to the transformation system. The condition of the data and transformation are checked for system consistency, if it is consistent the system is sent to the warehouse. After the data is processed and stored in the warehouse it is distributed to the end-user through the application layer which acts as a wrapper enclosing the application, enterprise warehouse and the ERBAC security component.

To summarize the advantages of the ERBAC component, the roles of the RBAC gives a hierarchical use of the data in the system but does not provide usage control over the data. There are also problems in maintaining single user in multiple sessions. The UCON component provides the decision component which is stronger than the RBAC. It checks for the pre-approvals which are the pre-conditions that need to be satisfied for safe execution on some data. It checks for the system condition for every change that is made on the data and stops if the data is going to make the system state inconsistent. The on-going approvals for the mutable attributes and other object attributes whose usages limits are extended on some particular scenario are checked by UCON. This pre-approval, on-going approval and system condition check provides a strong authentication and clear decision process. So ERBAC provides a single unified framework for RBAC and UCON to exist together and provide a strong administrative component and a clear decision making system.

3.4.3.4 Experiment and Challenges

The security component designed above has been implemented in a simulated Data warehouse. The front end of the system is designed using Java, back end is designed using oracle 10G XE, The simulated system is an inventory system, and it helps in displaying the data management using the ERBAC component. The roles and groups are defined in the database by the administrator. The rights are also managed within the database using a rights

attribute associated for each data. It clearly states the usage limit, owner for the data. If the limit needs to be changed, it requires approval from the owner of the data. In some cases the administrator can override the owner's rights if the data is inconsistent or might damage the consistency of other data. The rights of the data can be mentioned by the owner and has all the rights to grant or revoke the access whenever he wants. So even if the data is granted to other user and it is in use the data can be revoked from continues access. The administrator will ensure the stable condition of such on-going approval or on-going revocation.

Application simulated will generate scenarios where in it can show case the list of pre-approvals needed for the execution of a process and incase it needs an on-going approval it request for the approval to the administrator or the data owner. Apart from the rights that are mentioned in the database, there is a data policy manager encoded in xml format which acts as a data layer. It interacts between the application and the database and manages the rights of the data. The rights manager gives a list of pre-approvals needed for executing the process. The process continues until the on-going approvals are granted, there are some mutable attributes which can be loaded while the process is executed.

Here we discuss some of the challenges and issues faced during the implementation of such data warehouse in an Enterprise wide data Warehouse. The data rights and usage limits should be specified clearly. The Role and Rights should not conflict with each other. The management of mutable attributes increases the process time. The on-going approval increases the cost of the query. The process is slowed when there are some objects loaded during the course of the process. This can be solved to an extent if the course of the process and the attributes that are needed are known, a knowledge engine can be maintained which can ensure to load all the attributes at the pre-approval state.

3.5 Summary and Directions

This chapter has discussed secure data warehousing. We started with a discussion of a definition for a secure data warehouse, the technologies for a secure data warehouse, functions of a secure data warehouse, and issues on developing a secure data warehouse. Key concepts in secure data warehousing include developing a secure data model, security architecture, and access methods and index strategies. We then described the design and implementation of a secure data warehousing system based on the Extended RBAC model.

While some progress has been made on secure data warehousing and we are seeing commercial products incorporate some security features, there is still a lot to do. We need to develop ways to integrate security policies in building a warehouse. We also need a thorough investigation of the security issues in both building a warehouse as well as extracting data from the warehouse. We need to examine the security impact on integrating data mining with data warehousing. Finally we need to examine the inference problem and privacy problem that arise due to data warehousing and data mining. Some discussions on data mining, security, the inference problem and the privacy problem are given in [THUR02b].

REFERENCES

[BEST02] Bestougeff , H., et al, (Editors) Heterogeneous Information Exchange and Organizational Hubs, Kluwer, MA, 2002.

[INMO93] Inmon, W., “Building the Data Warehouse,” John Wiley and Sons, NY, 1993.

[PARK04] Jaehong Park and Ravi Sandhu. “The UCONABC Usage Control Model.” ACM Transactions on Information and System Security, Volume 7, Number 1, February 2004.

[SAND96] Ravi Sandhu, Edward Coyne, Hal Feinstein and Charles Youman, “Role-Based Access Control Models.” IEEE Computer, Volume 29, Number 2, February 1996.

[THUR91] Thuraisingham B., Secure Distributed Database Systems, Computers and Security, December 1991.

[THUR93a] Thuraisingham, B. Towards a Standard Multilevel Relational Data Model, Computer Standards and Interface Journal, 1993.

[THUR93b] Thuraisingham, B., Parallel Processing and Trusted Database Management Systems, Proceedings of the ACM Computer Science Conference, Indianapolis, IN, February 1993.

[THUR94] Thuraisingham, B., Security Issues for Federated Database Systems, Computers and Security, Volume 13, #6, 1994.

[THUR97] Thuraisingham, B., Data Warehousing, Data Mining and Security, Presented at the IFIP Database Security Conference, Como, Italy, 1996 (paper published in formal proceedings by Chapman and Hall, 1997).

[THUR05] B. Thuraisingham, Database and Applications Security, CRC Press, 2005

CHAPTER 4
ERBAC IN ENTERPRISE RESOURCE PLANNING SYSTEMS

Authors - Srinivasan Iyer and Dr.Bhavani Thuraisingham

The Department of Computer Science

The University of Texas at Dallas

P.O. Box 830688

Richardson, Texas 75083-0688

4.1 Abstract

Security in Enterprise Resource Planning ERP [1] system is a big concern because they integrate all the data and processes in an organization. There are some ERP systems which also provide service through internet like e-commerce where attacks are abundant. Many of the existing ERP systems use Role Based Access Control (RBAC) [2] for data security. There is various security issues linked with the existing ERP systems. This chapter mainly deals with the drawbacks of RBAC [2] in ERP systems. This chapter also proposes an Extended RBAC security component which is a combination of RBAC and Usage Control (Authorization, oBligation, Condition) UCON_{ABC} [3]. Though UCON gives a unified system framework it is not a substitute for the traditional access control lists. So this chapter takes the best of the both worlds and creates a new unified secured framework for all data and process integration.

4.2 Introduction

4.2.1 Enterprise Wide Systems (ERP) Overview

Enterprise Resource Planning (ERP) system helps an organization to integrate all its facets of a Business into a single unified system. The integration is mainly done using different resources like computer software and hardware of the organization. The ERP management is designed to plan the utilization of all the Enterprise wide resources to its optimal level. ERP has two main parts 1. ERP system manages processes like manufacturing, logistics, and distribution, inventory, shipping, invoicing and accounting for an organization. 2. ERP software controls business activities, like sales, marketing, delivery, billing, production, inventory management, quality management, and human resources management.

There is a general misconception that ERPs are back office systems which does not involve end users like clients, customers or general public but there are systems like customer relationship management (CRM) systems that deal directly with the customers, or the eBusiness systems such as eCommerce, eGovernment, eTelecom, and eFinance, or supplier relationship management (SRM)[5] systems. ERP is a unified system with cross functionality of operations and production of an enterprise. In addition to manufacturing, warehousing, logistics, and Information Technology, this would include accounting, human resources, marketing, and strategic management.

ERP systems are made of different business components. Security is an important

component of any ERP system. Many of the existing ERP systems use Role Based Access Control (**RBAC**) for data security. ERP is integration of all data in the organization with the business process, so the data needs security while being integrated with other data in the organization. Role based Access Control (**RBAC**) is one of the traditional access control technology that has been implemented based on the roles. This chapter mainly deals with the security component of the existing systems its advantages and disadvantages, Security issues that exist in ERP systems, Issues that exist in RBAC, Extension of RBAC with Usage Control UCON_{ABC}, Issues in extension, Implementation and Summary and Future Directions. The main objective of the chapter is about the extension of RBAC and UCON_{ABC} into an existing ERP system. UCON provides Unified framework for all digital resources.

4.3 History and Background of ERP

4.3.1 History of ERP

The history [6] of ERP can be traced back to the 1960's, when the focus of systems was mainly towards inventory control. Most of the systems software was designed to handle inventory based in traditional inventory concepts. The 1970's witnessed a shift of focus towards MRP [7] (Material Requirement Planning). This system helped in translating the master production schedule into requirements for individual units like sub assemblies, components and other raw material planning and procurement. This system was involved mainly in planning the raw material requirements.

Then, in 1980's came the concept of MRP-II (Manufacturing Resource Planning) which involved optimizing the entire plant production process. Though MRP-II, in the beginning was an extension of MRP to include shop floor and distribution management activities, during later years, MRP-II was further extended to include areas like Finance, Human Resource, Engineering, Project Management etc. This gave birth to ERP (Enterprise Resource Planning) which covered the cross-functional coordination and integration in support of the production process. The ERP as compared to its ancestors included the entire range of a company's activities.

However, it has been within the last five years that ERP has really taken off and seen record revenues by the software companies. In the past, ERP software was used to number crunch and schedule manufacturing processes. Management was not using ERP to its full

potential. Today, ERP is the foundation of businesses domestically and globally. It is used as a management tool and gives organizations a great competitive advantage.

4.3.2 Background of RBAC in ERP

CODA [8] (Complex Organic Distributed Architecture) is Role based Secured Business intelligence system for enterprise. CODA is an Object Oriented Layered Architecture. There are five layers in CODA.

1. *Operations*: This layer deals with simple linear data, which usually corresponds to typical transaction processing and business operations.

2. *Monitor operations*: In this layer, the data is often dimensional and aggregated. For instance, data is organized by time or group.

3. *Monitor the monitors*: This layer deals with multidimensional data and provides capability for analyzing trend behavior.

4. *Control*: This level should be able to “learn” about simple emergent behavior, trends and forecast and be able to run predictions and simulations automatically.

5. *Command*: This is the highest level, which has the ability to recognize new threats and opportunities and propose solutions.

Data sources, which can be data marts, data warehouses and legacy processing, must be attached to the right layer. The criteria to allocate a particular element to a certain layer (level) in the architecture are defined according to their processing type as in the VSM Viable Systems Model [9]. The Viable Systems Model is based on the way a biological organism, such as the human nervous system, processes data in terms of objectives. Incoming data is leveled according to the type of activity performed and filtered so that only the relevant information is presented when decisions are made.

The architecture of the security system supports a set of architectural properties, namely, scalability, effectiveness in ensuring secure access, and good performance, which are required for most enterprise security system. In addition to access control, the proposed architecture provides all the functionality required by a security system, including authentication and auditing.

4.4 ERP Security Issues

Existing ERP systems mostly use traditional security techniques, the security component

present in the present ERP systems use mandatory access control. One of such method is RBAC. There are various security issues that are involved in the ERP systems. Sensitive information cannot be displayed to unintended users. If it is disclosed to unintended users it may effect in information loss, misuse, or unauthorized access to or modification of, which could adversely affect the national interest or the conduct of federal programs, or the privacy to which individuals are entitled under *5 U.S.C. Section 552a* (the Privacy Act), but that has not been specifically authorized under criteria established by an Executive Order or an Act of Congress to be kept classified in the interest of national defense or foreign policy. (Systems that are not national security systems, but contain sensitive information, are to be protected in accordance with the requirements of the Computer Security Act of 1987 (P.L.100-235) [INFOSEC-99]

Critical functions only are performed by the right people in the organization. In case if the Critical functions are not performed by the right people then they might not have a profound knowledge of the scenario and may not be able to perform the function to its optimal. ERP system should give users access to all the relevant information, to make the optimum decisions. In order to avoid from disclosing the sensitive information from the unintended users the relevant information should not be concealed from rightful users and hinder the decisions. The scope of view to the authorized detail should be maximized.

E-Commerce requirements should have a security plan developed. The ERP systems are spread not only within organization. They are also available over the internet, this makes it more vulnerable and open to attacks. It has to develop a security plan such that it prevents attacks like denial of service, worms, viruses and millions of other illegitimate users who try to get access to the target information in order to misuse, or damage the information from further use. There should be appropriate security and control over your data. The system should comply with external and internal audit requirements. The system should eliminate the disclosure of confidential information. The system should know the reliability of the network.

4.5 RBAC in ERP

The stringent laws on privacy and the necessity to maintain confidentiality on financial institutions and other enterprise systems makes the identity management a critical factor and

so it needs a complex security model. The security model should be easy to implement, low in maintenance, should administrate access controls ensure network and data security. While RBAC can be challenging to design and implement, it can be tailored to a company's business model and security risk tolerance. Once implemented, it scales for growth and requires minimal maintenance

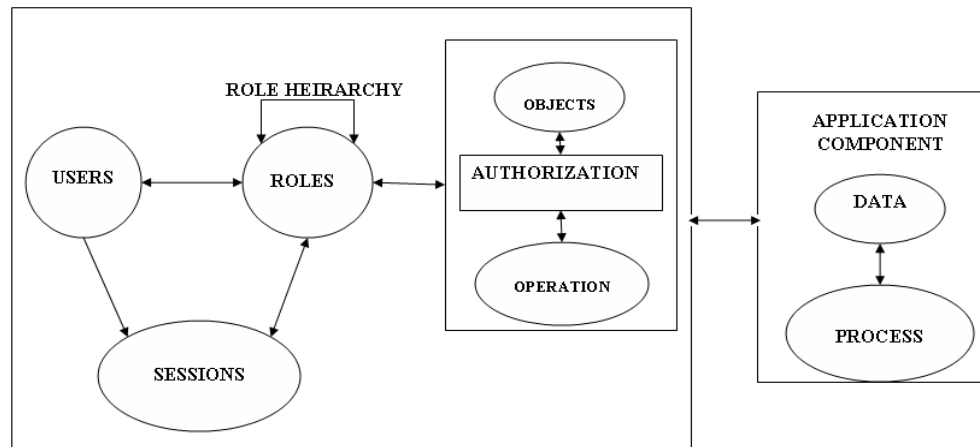


Figure 8. RBAC Component implemented in ERP Application

Once all of the employee roles are populated into the database, role-based rules are formulated and workflow engine modules are implemented. Through these elements, role-based privileges can be entered and updated quickly across multiple systems, platforms, applications and geographic locations RBAC provides company wide control process for managing resources. RBAC systems also can be designed to maximize operational performance and strategic business value. They can streamline and automate many transactions and business processes and provide users with the resources to perform their jobs better, faster and with greater personal responsibility. With an RBAC system in place, organizations are better positioned to meet their own statutory and regulatory requirements for privacy and confidentiality, which is crucial for health care organizations and financial institutions, as well as requirements imposed by external business partners and government agencies. Directors, managers and IT staffers are better able to monitor how data is being used and accessed, for the purpose of preparing more accurate planning and budget models based on real needs.

4.6 General Implementation of RBAC in ERP

4.6.1 Overview

RBAC is best implemented [10] by applying a detailed and structured framework that breaks down each task into its component parts. The system design is shown below. The components are shown as per implementation. The RBAC Security components are linked with the application of the enterprise system.

1. *Develop a detailed security plan:*

This process helps in easy implementation of the RBAC model. The RBAC security component should not be added at the end of the system implementation. The setup should be done along with the implementation of the actual enterprise systems. System design should consist of security along with other process components.

2. *Setup hardware and software requirements*

The hardware and software requirements like network router, firewall should be set respectively. This step calls for identification and listing of all servers, databases and applications. Only then can business units and management determine the level of security required for each application and data source, based on the core mission, the level of security and/or confidentiality desired, and the need for regulatory or statutory compliance.

3. *Define Users/Groups*

As shown Figure 8 Users of the system can be human beings or software components. User groups are created to manage the assignment of users to roles, as different users may acquire the same authorizations.

4. *Define Permissions*

Permission is an approval of a particular mode of access to one or more objects in the system. In general permission is an operation on an object.

5. *Define Sessions*

As shown Figure 8 Session is a dynamical mapping of a user to a role. In case if the application needs customization then there is a flexibility that can be implemented in the session mapping one user to many roles and many users to many roles.

6. Policy, Roles and corresponding Access Control

The Policy support defined for the corresponding applications gives the role hierarchy, role constraints their mapping to the user in each session if it is many to many or many to one. It also gives the corresponding access control for the respective roles.

7. Define Administration Security

The Administrator is the key person. He knows about the security architecture of the system and knows the policy, constraints, users and their roles in sessions, Access authorities. He defines the set of administration security which will take care of the above operations.

8. Integrate RBAC across all applications

This step is one of the key steps in the security model of the enterprise systems. The integration of data and process applications along with the security component is the final and foremost step which will ensure that the application and processes are secured.

4.7 Issues of RBAC in ERP

Though RBAC has been implemented many modern enterprise systems, there are various 'issues in RBAC' [6] that makes it not an appropriate security model for all ERP systems. The issues are what is role and how is it going to be different from groups? In case of group users, how the access controls is different based on the data? Duties are not a part of traditional access control. RBAC is a traditional access control model. It does not take into account the operations that are made by the entities. RBAC does not mind about the permission that is required to be given or denied for the access of objects based on the operations made by the entities (users) in the system.

The Roles in an organization arise naturally and the access control depends on roles. The scope of RBAC in enterprise wide roles is limited. There are situation where a single user can take multiple roles in the same session. This kind of flexibility is needed and in case of RBAC it has to be planned at the time of security component implementation.

There is another scenario where a single user trying to enter into multiple sessions simultaneously. In such instances the user should be given flexibility to do so if he has the privileges and can be given access to all the level in each session that he is entitled to.

RBAC should be able to enforce access controls to the same user with multiple roles. The change in role should not allow him to hinder the work that has already been done.

Temporal dependencies arise in organizations where order of causality should be maintained. In RBAC the decision factors depend on the access control at the time of requests rather than the on-going control which needs relatively long access. RBAC does not consider mutability.

4.8 UCON Overview

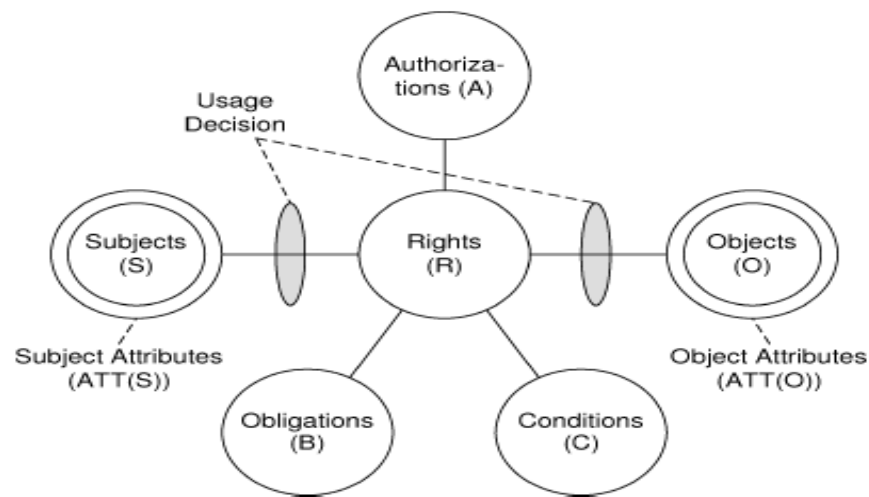


Figure 9. UCON_{ABC} Model Component

Usage Control is one of the modern approaches to provide security and access control model to data and its entities. It is an enriched and refined access control which covers the traditional access control list. UCON has a clear definition of access control discipline and knows the scope of the security. Usage Control integrates Authorizations **A**, Obligation **B** and Conditions **C**.

UCON_{ABC} consists of eight core components shown in Figure 9 [3]. They are subjects, subject attributes, Object, Object Attributes, Rights and the other three user decision components Authorization, Obligation, Conditions. Traditional models use only authorization for decision process. Obligation and Conditions are new concept defined in usage control which rectifies some of the decision process with the temporal dependencies.

A subject is an entity associated with certain attributes which has certain rights on objects. Authorizations checks if the subject has right over an object. It checks for subject attributes, object attributes and set of rights for authorization taking into account even the ongoing authorizations. Obligations are a functional predicate that has to be verified before a subject exercises usage on a specific attribute. The Pre-decision process needs Pre Obligation (Pre B) Approvals. The decision process which is on going needs to be revoked or continued needs On-going Obligation (On B) approval. Conditions are system oriented or environmental decision factors. They are not similar to authorization or obligations. They do not have direct relation with either subject or object and their attributes, they depends on environment.

4.9 Extended Role based Access Control

4.9.1 ERBAC Overview

The existing systems mostly use RBAC which has more limitation with respect to resource management. It has many issues regarding decision process, multiple roles, multiple session and many other temporal dependencies. So the enterprise needs are not fully met only having RBAC has a security model. UCON is one of the modern security models which covers most of the traditional access model functionalities and has more new functions. UCON cannot exist alone in an enterprise and manage all resource. It is not one for all complete solution. UCON has to co exist with some other traditional component in order to provide a strong secured enterprise wide system because it's a specific component which is mainly strong only in decision factor. The delegations of role are derived from traditional access control lists. The administrative functions are not as robust as in RBAC. So RBAC and UCON combination will form a strong access and usage control security component. This newly proposed component is E-RBAC (Extended Role Based Access Control)

4.9.2 ERBAC System Architecture

The system architecture of the Extended RBAC is similar to that of the existing security model. It is going to be a combination of the RBAC security component and UCON security model. The Administrative Security, Role delegations are part of RBAC component and the rights of objects and decision process is a combination of UCON. It will take into account the obligation approvals for pre and on-going transactions. It will also check the environment

conditions before it gives the approval for authorization and the predicates of the authorization the obligations. The UCON model also helps in tracking the temporal dependencies there by helping to know if the current system can grant or revoke the operation that is going to be performed or that is already being performed. This component is not available in the existing role based access control. Combining both the models we get a secured system which encompasses both a highly secured administration rules. It also helps in the protection of identity management. The rights manager helps in checking the privacy constraints of the corresponding data.

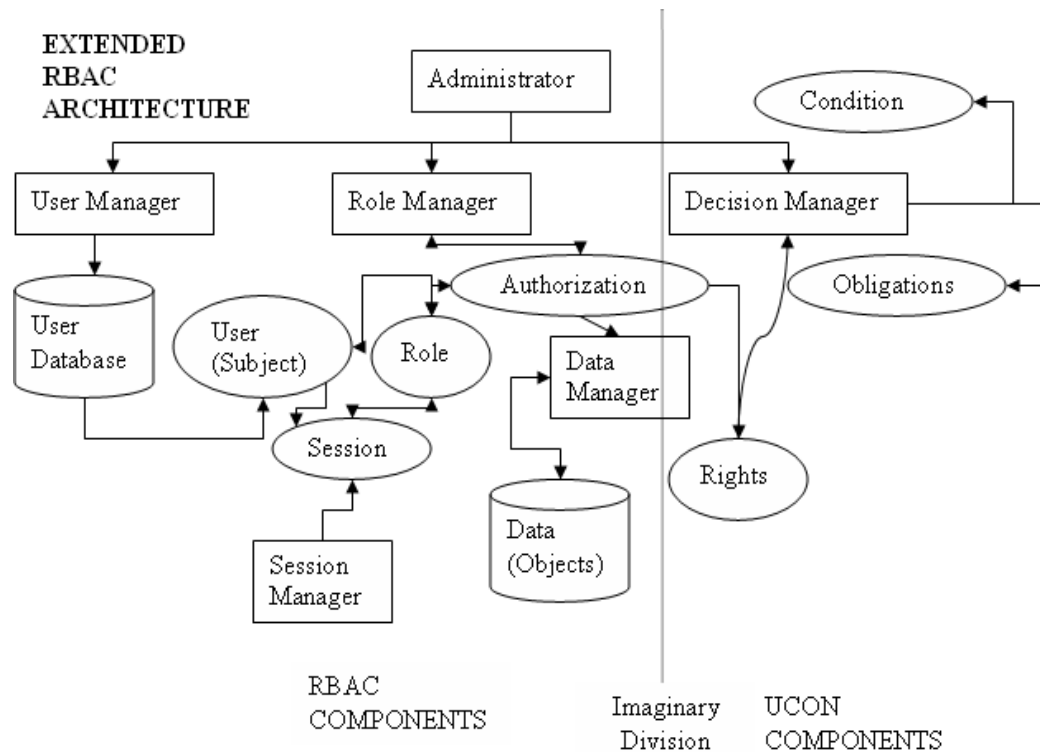


Figure 6. E-RBAC with UCON Extension

In the above shown figure 6 we see the architecture of the newly proposed system RBAC with UCON. The Administrator component provides the extension of role based access and usage control. The architecture also shows the imaginary division between the RBAC and UCON components. There are five managing components as shown in Figure 3 User Manager, Role Manager, Decision Manager, Session Manager and Data Manager. The administrator is the configuration controller which manages all the security components.

User manager takes care of the list of users in the database using the system. It interacts with the role manager and gets the corresponding role of the user for the respective session. The session manager helps in maintaining multiple sessions and maps the history of the user in each session. Each time the user requests for some data the UCON comes into effect. The decision manager comes into the context and checks for the condition of the current system and check if it is going to be consistent even after the request being served. The Obligations are checked and the rights for the corresponding data are checked for pre approval and on-going approval. When the predicate approval is done depending on the role of the user the authorization is done. If all these decision process are checked then the user request is serviced. Thus the role based access and usage control is setup in a same system which has a strong administrative and decision process with temporal dependencies, mutability and identity management.

4.10 ERBAC in ERP System Architecture

4.10.1 Security Based Architecture Overview

Enterprise wide Resource Planning system is by itself very complex because it needs to integrate all the data, process and business operations into single system. The security for such a complex system needs to be effective because the weak link of the secured system is also the strongest link. The application is spread across the enterprise which spreads over many geographical locations. So the security plan needs to foolproof to avoid illegitimate users from intruding or attacking the system resources.

Existing system use Role based access control since it is a traditional security component over 30+ years and has been efficient and easy maintenance. The disadvantages have been discussed in the previous section. The issues in RBAC, Its existing role in ERP, disadvantages of RBAC in ERP , UCON model which has an architecture that can support the decision process and give a time tracking for all the process which request some object.

The Application component is linked with the security component and makes use of the administrative security available in the component. The security is based on both the access control and usage control.

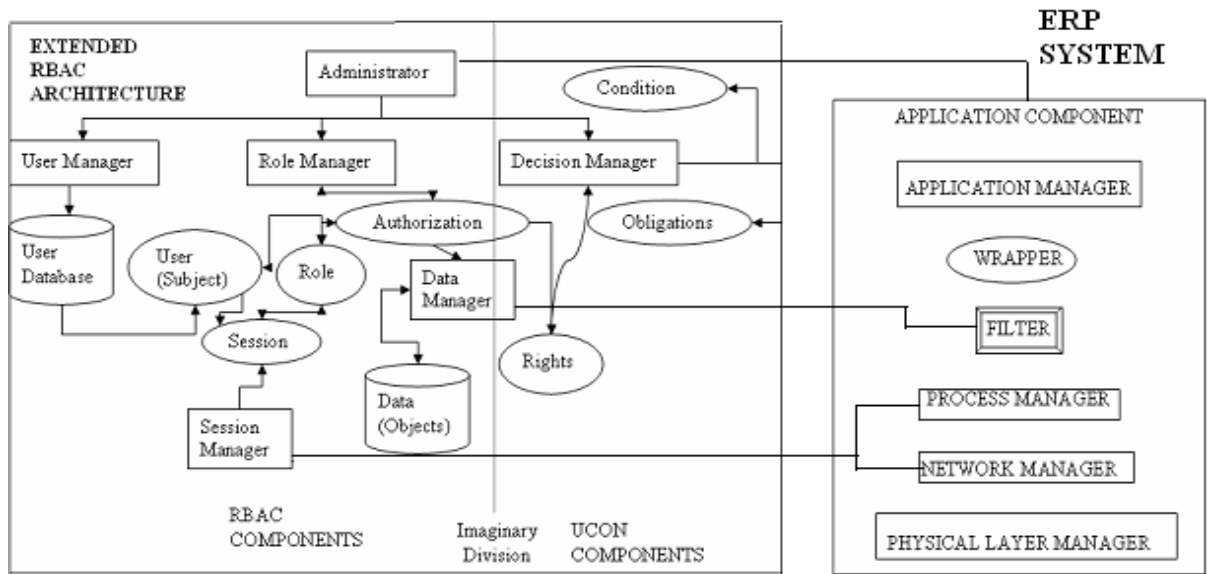


Figure 10. ERBAC System Architecture in ERP System

Administrative security

The administrator component is linked to three main components: User manager, Role manager, and decision manager. The user manager has a user database. The role manager helps in the mapping of the roles with the users with the corresponding users. In the ERP component, the session manager is controlled by the network manager and process manager. The network manager, with the help of the underlying network protocol, ensures that the session created is secured and does not allow any illegitimate users to log on to the system. The process manager maintains the list of sessions that can exist simultaneously without disturbing the consistency of the system. The Filter in the ERP component filters the data which needs to be processed before the integration. It ensures that there are no unprocessed data that enter the list of processed list. The Administrator is responsible for the user, role, and session relation and the authorization of the corresponding roles and the data are employed in the object database. The decision process entirely depends on the role and the usage of the data by any user. The user cannot get access for the data that he is not entitled to. The decision of any data to be granted, revoked, or continued depends on the condition and obligations and the authorization. There can be many constraints that can make the user get access denied for which he is entitled. This depends on the existing conditions of the system.

Policy Support

Role Delegation: Users or group of users are assigned to roles [11] via the Role Manager. The Role manager has access to the object database and can add or delete roles to/from a user's set of roles. Before adding new roles, the Role manger checks whether the new set of roles is consistent i.e. there is no separation of duty constraint within the new set.

Usage Constraints:

Rights: The rights in the ERBAC are of two types. The Role of the user has certain rights and the object has certain rights. The rights are managed by the decision manager and role manager is combination.

Decision Process: Once any user requests for a data. The Role manager has certain functionalities with which it checks if the role of the user has authority to access the data. Then follows the decision process where the temporal dependencies are taken into account and checked if the usage of the data can be allowed. The usage decision process checks for the condition component which checks for the environmental condition, all the processes, their current state and if the corresponding request is served, is it going to affect the consistency of the system, then if the conditions are cleared then the obligation which is a predicate for the authorization process. There are two types of obligation in all transactions [12]. The pre-obligation which will approve if there are no current dependencies that are executing in the system and the approval will not affect the system in anyway. The on-going obligation which checks if the current execution is stopped or revoked or continued after the approval of the request which is in queue will that affect the system? If those two predicates are approved then the authorization process for the corresponding to the request again has two pre-authorization which is similar to pre-obligation and on-going obligation. In such authorization the check is for the current role and rights of the user and data respectively.

Thus the above administrative component, policy, roles, usage and decision help in proper execution of the ERBAC system in an ERP environment.

4.11 Challenges in Implementing ERBAC in ERP

The security component designed above has been implemented in a simulated Data warehouse. The front end of the system is designed using Java, back end is designed using oracle 10G XE, The simulated system is an inventory system, and it helps in displaying the

data management using the ERBAC component. The roles and groups are defined in the database by the administrator. The rights are also managed within the database using a rights attribute associated for each data. It clearly states the usage limit, owner for the data. If the limit needs to be changed, it requires approval from the owner of the data. In some cases the administrator can override the owner's rights if the data is inconsistent or might damage the consistency of other data. The rights of the data can be mentioned by the owner and has all the rights to grant or revoke the access whenever he wants. So even if the data is granted to other user and it is in use the data can be revoked from continues access. The administrator will ensure the stable condition of such on-going approval or on-going revocation.

Application simulated will generate scenarios where in it can show case the list of pre-approvals needed for the execution of a process and incase it needs an on-going approval it request for the approval to the administrator or the data owner. Apart from the rights that are mentioned in the database, there is a data policy manager encoded in xml format which acts as a data layer. It interacts between the application and the database and manages the rights of the data. The rights manager gives a list of pre-approvals needed for executing the process. The process continues until the on-going approvals are granted, there are some mutable attributes which can be loaded while the process is executed.

Here we discuss some of the challenges and issues faced during the implementation of such data warehouse in an Enterprise wide data Warehouse. The data rights and usage limits should be specified clearly. The Role and Rights should not conflict with each other. The management of mutable attributes increases the process time. The on-going approval increases the cost of the query. The process is slowed when there are some objects loaded during the course of the process. This can be solved to an extent if the course of the process and the attributes that are needed are known, a knowledge engine can be maintained which can ensure to load all the attributes at the pre-approval state.

REFERENCES

1. Enterprise Resource Planning: Our experience and learning in ERP implementation Rakesh Agarwal, Arup Ratan Raha, Bhaskar Ghosh ERP Implementation in state Government. Watson Ed. Using a case study to test the role of three key social enablers in ERP Suprateek Sarkar, Allen S. Lee ERP Implementation planning and Structure. Wayne Brown October 2004.

2. SANDHU, R., COYNE, E., FEINSTEIN, H., AND YOUMAN, C. 1996. Role-based access control models. IEEE Computer (Feb.), 38–47. Proposed NIST Standard Role Based Access Control Model David F. Ferraiolo, Ravi Sandhu, Serban Gavrila, D. Richard Kuhn, Ramaswamy Chandramouli

3. UCON: Jaehong Park, Ravi S. Sandhu: The UCONABC usage control model. ACM Trans. Inf. Syst. Secur. 7(1): 128-174 (2004)

Figure 9 [3] UCON: Jaehong Park, Ravi S. Sandhu: The UCONABC usage control model. ACM Trans. Inf. Syst. Secur. 7(1): 128-174 (2004)

4. Issues in RBAC Ravi Sandhu.

5. Organizational engineering (OE): Supplier network management: evaluating and rating of strategic supply networks Nikolaus Muessigmann, Antonia Albani April 2006 Proceedings of the 2006 ACM symposium on Applied computing SAC'06

6. History of ERP http://www.ebz-eratungszentrum.de/pps_seiten/sonstiges/erp_engl.htm

7. Manufacturing applications: Simulation test bed for manufacturing analysis: a simulation test bed for producton and supply chain modeling S. T. Enns, Pattita Suwanruji Decemeber 2003 *Proceedings of the 35th conference on Winter simulation: driving innovation*

8. A Role-based Security Architecture for Business Intelligence S. Megaache, T. Karran, G. R. Ribeiro Justo Cavendish School of Computer Science, University of Westminster

9. S.Beer, "The viable system model: Its provenance, development, methodology and pathology", Journal of Operational Research Society, 1984, 35(1) pp.7-25.
10. Implementation Overview of Role based Access control for Business EnterpriseTrey Guerin and Richard Lord, Network Security Consulting
11. R. Sandhu, "Role activation hierarchies", In 2nd ACM workshop on RBAC, October 1998.
12. SCHAAD, A. ANDMOFFETT, J. 2002. Delegation of obligation. Proceedings of the Workshop on Policies for Distributed Systems and Networks.

CHAPTER 5

CONCLUSION

My research towards my Thesis is given clearly in three different chapters. The chapter two clearly illustrates the trust management techniques with confidentiality for a coalition data sharing environment. It also gives the experimental results and future directions in which the study of negative message relation among agents with trust variations is an important future direction that can be pursued upon. The third chapter gives a detail design steps of the data warehouses, The extended design, its advantages and implementation issues, it also clearly gives what are the optimization that can be done in the future with respect to the extended Role based access control component. In third chapter the same security component is used in an Enterprise Wide Environment and checked for the issues in the existing systems, advantages of the extended design which combines best of the both worlds of usage control and Role based access control.

VITA

Srinivasan Iyer was born in Chengalpattu, India, on June 20, 1982, the son of Capt. R. Venkat and Vasugi Ramanan. He received his Bachelor of Computer Science and Engineering from Bharathiyar University, Coimbatore, India in 2004; in 2004 he worked for a year as a Software Engineer at Infosys technologies Pvt. Ltd., Bangalore, India. In August 2005, He entered the Graduate School of The University of Texas at Dallas.