Secure Multi-Agent System using Role Based Access Control in a Distributed Environment

P.Marikkannu, J.J. Adri Jovin
Department of IT
Anna University of Technology
Coimbatore, India

T.Purusothaman
Department of CSE & IT
Government College of Technology
Coimbatore, India

R.Baskaran
Department of Computer Science
Anna University
Chennai, India

ABSTRACT
Multi-Agent Systems are helpful in gathering and processing information about customers or users of a particular domain by means of efficient communication techniques between agents for an effective sharing of resources. Extraction of relevant data from enormous volume of data has become a tedious task in today's world. Multi-Agent systems reduce the complexity of the task by assisting the users to obtain the relevant information from various data sources. A multi agent system that fills the gap between a customer and resource provider through agent communications in the banking domain is proposed which has a service requestor module which can access the services offered by the Multi-Agent by making request with Interface Agents. Also the Multi-Agent system is made up of a number of role based agents which provide services such as user authorization, negotiation, accessing and extracts the relevant information from databases to present them to the users. During the information access process, the role agent negotiates with the storage repository about the user access. After successful validation, the role based multi-agent system provides information to the user. The Multi-Agent System provides all the bank facilities to its customers when their authentications match, including viewing account information, performing transfer, viewing transactions and the branch information.

General Terms
Performance, Design, Experimentation, Security.

Keywords
Agent Communication Language, Java Agent Development Framework (JADE), Multi-Agent System (MAS), Role Based Access Control (RBAC).

1. INTRODUCTION
In this work, a Role Based Multi-Agent System for providing effective and secure Bank transaction services has been projected In addition it helps the bank employees and other concerned users to access information about customers. The Multi-agent system discussed in this work includes a number of distinct and expedient features such as ease of use, effective communication between customers and service providers, segregation from resource specific details, deviations in the system and internal complexity. The main focus of this work is on the design and implementation of a role based Multi-Agent System, which offers a wholesome access to information present in the database. The Multi-Agent System provides functionalities like User Authentication, Service Registration, and identification of Role-based Agent. It also uses Negotiation agent to access and trust negotiations, Database agent and Interface Agent to accept and display request service results.

The Role Based Multi-Agent System developed in this work encompasses Agents which are developed using Java Agent DEvelopment Framework (JADE) and Java APIs. Moreover, this system provides additional features such as the Login Validation, Service Registration, Role Agent Identification, and User Interface for an effective interaction between customer and the system. The role agent retrieves the information from database and sends it to the customer through an interface agent.

The main contributions are the establishment of a multi agent system to improve the security of the system using access control techniques, effective communication for fast retrieval and performance enhancement, and increased consistency using multi agents.

The remainder work is organized as follows: Section 2 provides a review of related works and compares them with the multi-agent system proposed in this work. Section 3 portrays the architecture of the multi-agent system developed in this research work. Section 4 explains the protocols proposed and used for effective agent communication. Section 5 discusses the implementation details of this system and also the results obtained from this work. Section 6 gives the conclusion on this work and suggests some possible future enhancements.

2. RELATED WORKS
Research on Information sharing and Trust negotiation using intelligent agents is gaining significance in the recent years due the growing necessity for an effective decision making using agents.

There are numerous works related to agents, trust negotiation and access control that exist in the literature. Aurora Vizcanaino et al [1] proposed a multi-agent model which is suitable for developing a generalized knowledge management system. However, it is essential to develop a Multi-Agent System separately for crucial applications such as Banking, in which access control and trust negotiations play a vital role.

The basic idea is that an explicit communication action will deserve a cost and suppose the global reward function of the agent team. The communication cost and rewards are known. The intelligent software agents are used to share information with confidentiality and trust [2]. It clearly defines an Intelligent Software Agent background of Information sharing in Intelligent Agents and the trust in the agents. The information that is shared necessitates security policy enforced based on the domain of the information and trust level of individual agent. Software agents [3] can perform tedious, repetitive, time consuming, and analytically complex tasks more accurately and reliably than people. They can serve as expert assistants in monitoring, troubleshooting, and predicting failures in complex processes.

C.VGoldman et.al [4] put forward a decentralized collaborative multi-agent communication model and mechanism design based on Markov Decision Process (MDP) which assumed that agents are full-synchronized when they start functioning, but no specific optimal algorithm is presented. Also, there are no experimental results showing their algorithm can work on large teams. Incomplete information theory is another way to solve the information sharing problems [5].

Fabio Belliemine [7] 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. It uses a probability method to coordinate agent team without explicit communication by observing teammates' action and coordinating their activities via individual and group plan inference.

3. SYSTEM ARCHITECTURE

The Role Based Multi-Agent System accepts the customer request via interface agent. The facilitator is used to register the service of role agents. The role agent has the capability of forwarding the request to negotiator for the validation of request and account details. The database agent possesses the ability to retrieve the customer information. The interface agent is used to display the customer details. The entire system architecture is shown in the Fig 1.

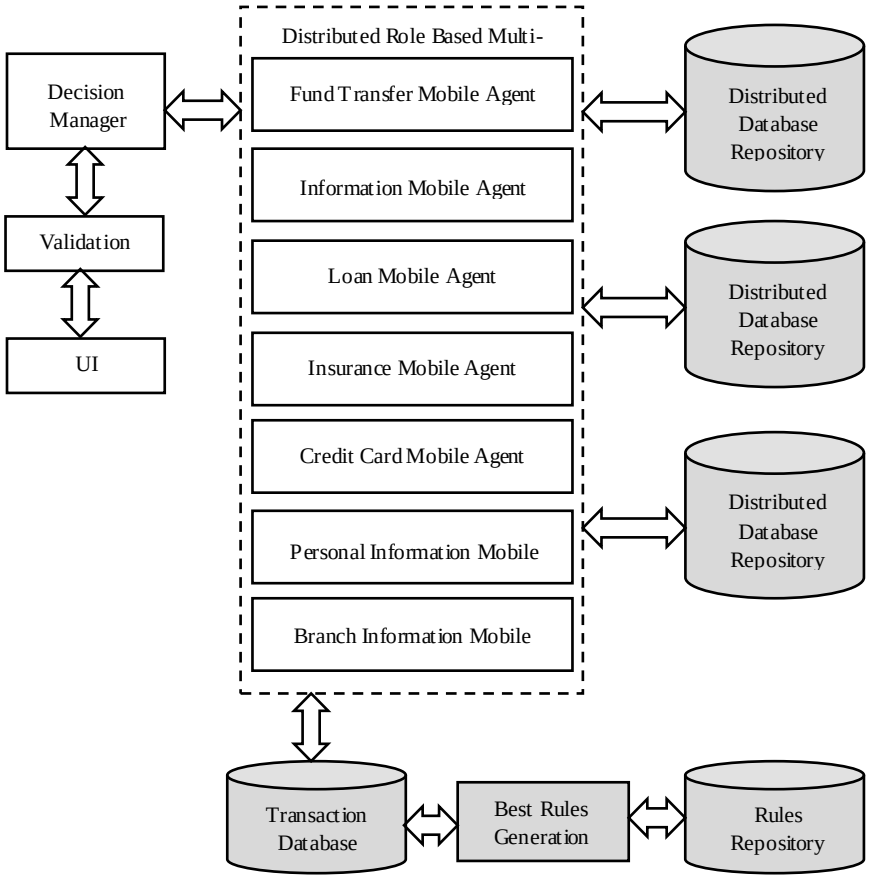

Fig 1: Role-Based Multi Agent Architecture

3.1 User Interface

Graphical User Interface accepts the username and password of the customer. The security manager validates the username and password of the customer. It sends the authenticated information to the Role Based Multi-Agent.

3.2 Mobile Agent Creation

In the Mobile Agent Creation module, the facilitator acts as an administrator agent to process the customer request. The Distributed Role Based Multi-Agents are usually created against

the scope of the request. The server with appropriate resource is identified to process the customer request. The facilitator coordinates the mobile agent to process customer request. The facilitator registers the agent to access a particular service using its Agent ID. After completion of the customer service the facilitator deregisters the service i.e. terminates the agent activities.

3.3 Information Mobile Agent

The Information Mobile Agent accepts the customer username and evaluates it with the usernames existing in the database. The customer request is obtained by the interface agent from the customer agent .The interface agent then sends this to facilitator to identify the role based mobile agent. The role based mobile agent to process the request posted by the customer. Role Based Mobile Agent will make the request to database agent for retrieving the customer information.

3.4 Loan Mobile Agent

The user is validated and the system lists the available bank loan schemes to the customer. Before approving the required loan to the customer, the Loan Mobile Agent checks the customer's history. The customer information is negotiated and if the customer has sufficient balance, the loan is sanctioned. New loan is not permitted unless the customer pays the previous dues. The customer balance amount gets updated when permitted to avail the loan.

3.5 Credit Card Mobile Agent

The Credit Card Agent provides access rights to the customer after validation. The credit card agent sends the customer information to interface agent. The interface agent directs the Credit Card Agent to register its service with the facilitator. The role of the agent is to process the customer request. The role based agent checks the customer's earlier due concerning cash or purchase. If customer do not have due then the customer is allowed to use the credit card within the permitted limit. If the customer has dues, credit card payment is barred.

3.6 Insurance Mobile Agent

The Insurance agent delivers information to all customers about the insurance offers (i.e. access to polices, brochures and price list). A foremost task of the Insurance Agent is to describe the features of different types of insurance and their aspects. The idea of providing insurance services is shown in Fig 2.

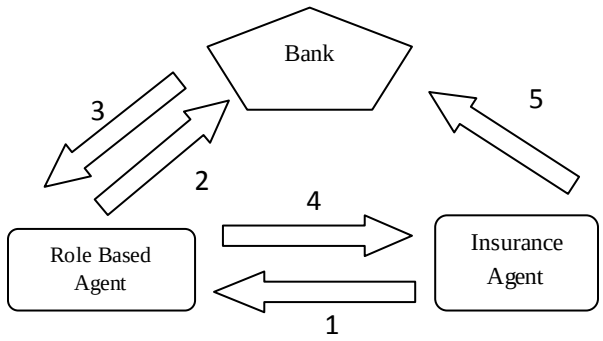

Fig 2: Intra Bank Money Transfer

It shows the different scenarios which have been implemented with the main objective of evaluating the feasibility and effectiveness of automated agent-based transactions [8].

1. Electronic bill
2. Payment order
3. Electronic notification
4. Electronic notification
5. Verification

Role based Agent specifies an insurance policy with Insurance provider Agent for $2500. It orders the bank agent to transfer $2500 from customer's bank account to Insurance provider's Account. In this case both the person has the account in the same bank.

1. Role based Agent to Insurance provider agent negotiation: The negotiation is successful if the two agents find a common agreement about insurance policy and its price. Depending on the specific policy, the final agreement can include different information and specification. In this kind of bilateral negotiation, one agent plays the role of buyer and the other agent plays the role of seller. If the negotiation terminates with an agreement, the Role agent must pay the insurance agent and the seller must provide service to the buyer. Depending on the type of payment that has been negotiated, explicit interactions with the bank agent may be required.

2. The Role agent makes the payment and gets the necessary insurance agent's bank account information and asks its bank agent to transfer the amount of money specified by the Insurance agent account.

3. The Bank Agent verifies whether Role Agent has sufficient money to make the payment. In such a condition, the Bank Agent transfers the specified amount of money from the buyer's account to the seller's account. If buyer doesn't have enough money in his account it should be notified.

4. The Insurance Agent provides the service to the Role Agent. The service delivery can be subordinated to complete the payment.

5. The Insurance Agent finally interacts with bank agent to verify whether the money transfer from the account was made.

3.7 Fund Transfer Mobile Agent

The Fund Transfer Mobile Agent accepts the sender customer username, the amount to be transferred and the receiver username and verifies it with the existing user credentials in the database. The sender customer bank balance is checked before transferring the fund to the receiver account. The transaction is successful if and only if enough balance is available else, it rollbacks the entire transaction. The Interface Agent receives the status of the transaction and forwards it to the customer.

3.8 Personal Information Mobile Agent

The Personal Information Mobile Agent accepts the customer username and verifies it with the database agent. The customer request is accepted by the interface agent and forwarded to the

agent facilitator for further processing. The Agent Facilitator creates the mobile agent and gives the direction of selected Distributed Database Server. The retrieved customer information is encrypted before moving to the origin host. After reaching the origin host the customer encrypted data is decrypted and then forwarded to Interface Agent.

3.9 Branch Information Mobile Agent

The Branch Mobile Agent accepts the branch code verifies it with database agent. The branch information request is accepted by the interface agent and forwarded to the agent facilitator. The Agent Facilitator creates the mobile agent and gives the direction to migrate to the appropriate Distributed Database Server. The retrieved branch details are encrypted before move to the host server. The branch information is secured before transferring the branch information to the interface agent.

3.10 Decision Manager

Decision Manager makes the decision from the role based multi-agents with the coordination of a data warehouse.

3.11 Rules Generator Agent

In the Rules Generating Agent, the completed processes are stored in the transaction database. The frequent Role based mobile agent sets are identified by using support and confidence. After obtaining the Role based agent sets, it can be associated with other Role based Agents for making the knowledge. The Rule Generation process uses the minimum confidence threshold (MCT) and minimum support threshold (MST) as input. After every transaction, the results of completed transactions are stored in the database. The frequent Role based mobile agents are identified by using Apriori Algorithm with the help of minimum support and confidence. This algorithm generates the candidate item sets and prunes the sets which are less than support and confidence level. The best rules are generated by making associations with the frequent role based mobile agents. The Association Rule will give the knowledge to the management for efficient decision making and also to improve the performance of the organization. The best rules are stored in the database for further performance improvement.

4. Multi-Agent Communication

In order to achieve the individual objectives or dynamically organize the actions, software agents need to cooperate with one another. Communication [6] is at the basis of agents interactions, ranging from the ability to exchange comprehensible messages (semantic interoperation) through traditional requests through which a particular action is performed (i.e., client-server approach), and the ability to dynamically organize and negotiate their behaviors.

4.1 Foundation for Intelligent Physical Agent (FIPA) specifies standard components that can be used [7]:

i. At the conversation level, the valid sequences of messages regulating specific interactions need to be determined. These valid sequences are called Interaction Protocols (INP) and they have to be known by all the interacting entities (i.e., transmitter and receiver/s).

ii. At the message or communicative act level, a shared Agent Communication Language (ACL) is being used. This enables the attitudes regarding the content of exchange to be expressed. The ACL structure establishes, for instance, whether the content of the communication is, for instance, an assertion, a request or some form of query. The two most known and deployed ACL in the world of agents are FIPA-ACL and Knowledge Query Manipulation Language (KQML).

iii. At the content expression level, the content of the message (i.e., a description of a partial world state) need to be described. This may contain references to objects, actions and functions in a given domain. Some of the existing Content Languages (CLs) used for this purpose are for instance FIPA-SL, Knowledge Interchange Format (KIF), etc.

iv. At the ontology level, a common approach is used to define a vocabulary and a set of agreed definitions is used to describe a specific domain in terms of objects, actions and functions.

4.2 Interaction Protocols for Agents Conversations

For synchronized interactions, ongoing agent conversations need to follow certain typical patterns. This means that certain message sequences are expected and at any point in the conversation, other messages are expected to follow. These typical patterns of message exchange are called interaction protocols and they represent the highest abstraction component in the Agent Communication Stack. In our example, when the Interface agent contacts the Role Based Agent for requesting the information of customer account, the standard FIPA-request protocol is used.

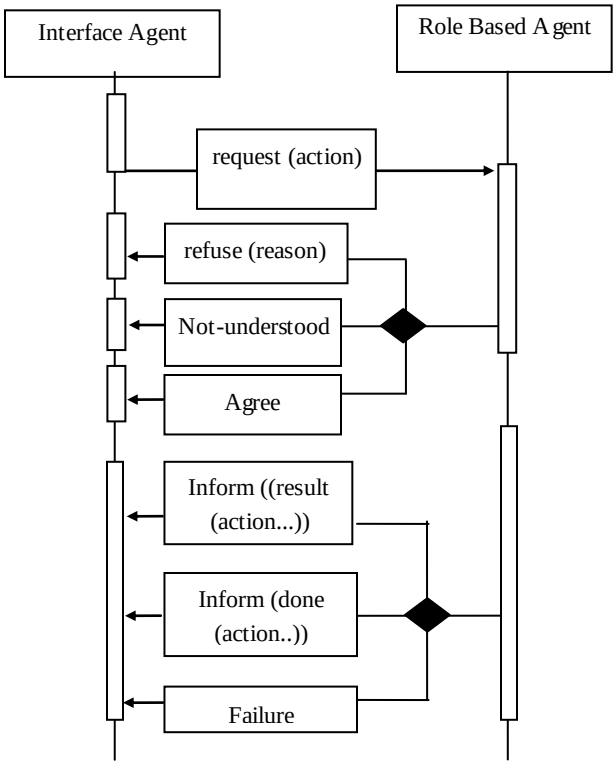

Fig 3: The standard FIPA-request interaction protocol

This identifies the specific kind of ACL messages which the agents are expected to exchange (i.e., the valid per formatives are request, refuse, not-understood, failure, inform) as shown in Fig 3.

The Interface Agent requests account -info to perform this by using the 'action' interface. The specified action together with the additional parameters to represent the content expression request is shown in Fig 4.

```
(request
    : sender(agent-identifier: name Interfaceagent)
    : receiver(agent-identifier:name negotiateagent)
    : language FIPA-SL
    : ontology(bank-ontology)
    : protocol(fipa-request)
    : content
      ((action(agent-identifier:negotiateagent)
         (account-info (accountno)(balance)(branch))
      )) )
```

Fig 4: The message sent by Interface agent for requesting agent Role Based Agent to retrieve the customer account information

5. EXPERIMENTATION AND RESULTS

The experimentation was performed in a group of systems with 2.26 GHz Intel Pentium IV systems with 512 MB RAM and 80 GB Hard Disk operating with Windows XP Operating System. The System is implemented with JADE 3.3, Java and Oracle 10g. Java uses its database driver called JDBC (Java Database Connectivity) to connect the database. The interface between Oracle and Java is provided by JDBC-ODBC protocol. A Bank application is being implemented with Multi-Agent Role Based Access Control in distributed environment. The results are shown from the Fig 5 to Fig 10, from which the following observations can be made: The processing time consumed is very less with respect to the number of transactions for Distributed Multi-Agent System which uses agents like Account Information Agent, Loan Agent, Credit Agent and Insurance Agent when compared with Client Server Systems. Knowledge can be obtained from the rules generator agent. Users can interact with this system using the Interface Agent. Users need to write a Java class that specifies some roles and actions like accessing customer information from database. User can launch the agent in two ways: (1) via GUI agent Wizard, (2) via command line.

5.1 Account Information Agent

In this module, the role based multi-agent GUI is created using JADE and Swing. The Account information agent gets the username through the interface agent. Role of agent is registered with facilitator. Now the user details are negotiated based on the behavior of the agent. Based on the agreement the account number and customer account balance from the account table and branch name from the branch table are joined before displaying the information.

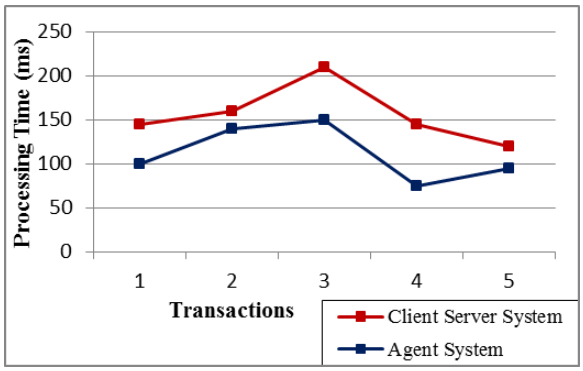

Fig 5: Comparison of Client Server System and Distributed Agent System for Account Information Agent

5.2 Loan Agent

In this module, authentication is checked and lists the available bank loan schemes to the customers. The user can select the options. Before sanctioning the loan to the customer, this checks the customer dues. The customer information is negotiated and if the customer has sufficient balance, the loan is sanctioned. New loan is not entitled unless the previous dues are paid. The customer balance is updated when permitted to avail the loan.

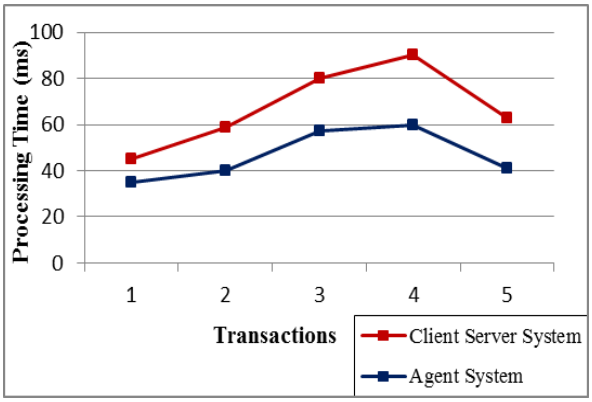

Fig 6: Comparison of Client Server System and Distributed Agent System for Loan Agent

5.3 Credit Card Agent

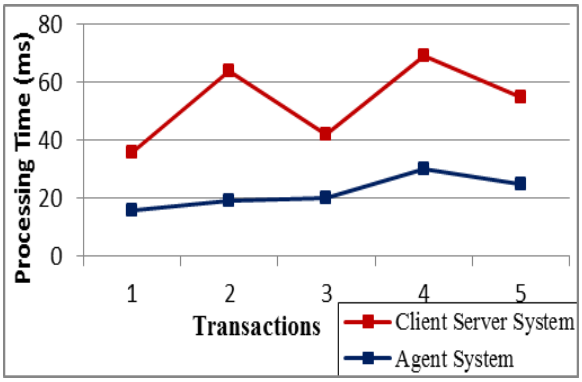

Fig 7: Comparison of Client Server System and Distributed Agent System for Credit Card Agent

In this module, the customer receives access rights after authentication. The credit card agent sends the customer information to interface agent. The interface agent coordinates to register its service with the facilitator. The role based agent checks the customer history. If customer does not have any due the usage of credit card is permitted to a limit. The implementation possesses a threshold value for cash and purchase value. The customer can purchase 100% of their limit values but cash credit is permitted to 95% only. Before sanctioning the credit cash or purchase the dues are checked. Customers with dues are not permitted to transactions using credit card.

5.4 Insurance Agent

In this module, the GUI will display the available Insurance details with coverage options. Various scenarios are implemented with the purpose of evaluating the feasibility and the effectiveness of automated agent-based transactions.

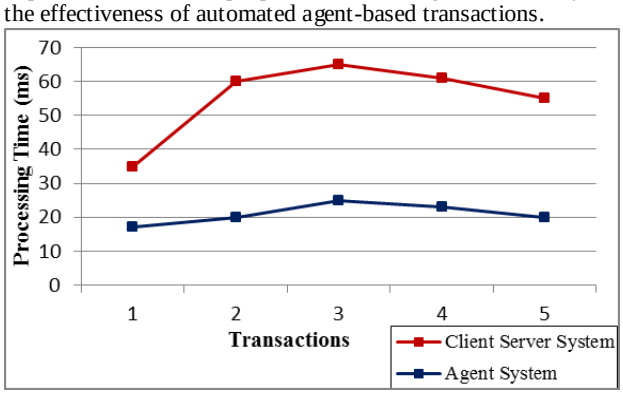

Fig 8: Comparison of Client Server System and Distributed Agent System for Insurance Agent.

5.4.1 Intra-Bank payment order

In this implementation, both the Role agent and Insurance provider agent has the account in the same bank. The insurance amount is deducted from the buyer's account and credited to the seller's account. If the buyer doesn't have sufficient balance, it can be informed.

5.4.2 Inter-Bank payment order

Consider a person Y. Y's Role based agent specifies an insurance policy with Insurance service provider agent – X for $2500. Y's Role Based Agent orders the bank to transfer $2500 from its bank account to the Insurance provider agent – X's account. In such a condition, the insurance provider account is managed by a different bank agent.

5.4.3 Credit Card Based Payment Scenario

The Role based agent stipulates an insurance policy with insurance provider agent for $2500. Role based agent pays the specified amount of money via customer's credit card account.

1. Role Based Agent to Insurance Provider Agent Negotiation: A successful negotiation terminates with an agreement about a specific insurance policy and its price. The customer's Role Based Agent asks to pay by using Y's credit card account (i.e. issuer bank). The Role based agent and Insurance provider agent first negotiate on the type of credit card that is accepted as payment. Based on the Insurance Provider Agent's payment infrastructure it accepts specific cards. The Customer's Role based agent securely submits the credit card information to the insurance provider agent.

2. Insurance Provider Agent to Bank Agent Interaction: The payment can be done in different ways. It depends on the type of agreement between the Insurance provider agent and Bank agent (acquiring bank) to which it had previously stipulated. In general, the insurance provider agent submits customer's payment data to the Bank agent. This does the credit card verification and provides an electronic receipt, which is forwarded by the Insurance provider agent to the Role based agent. If the credit card information is invalid the message that is sent back to the Role based agent is a failure message explaining the failure reasons.

3. The Bank agent reports the completion of the transaction to its customers. The Role based agent also receives an electronic receipt from the Insurance provider agent.

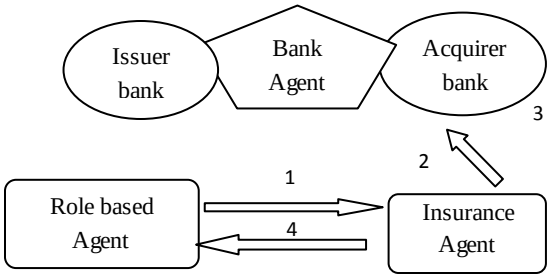

Fig 9: Credit card payment scenario

1. Payment order 2. Electronic receipt
3. Electronic receipt 4. Payment order

5.5 Rules Generator Agent

In the Rules Generating Agent module, the Completed processes are stored in the transaction database. The frequent Role based mobile agent sets are identified by using support and confidence.

RuleNo	RuleHead	Symbol	RuleBody
1	fnd	==>	acc
2	acc	==>	fnd
3	pr	==>	fnd,acc
4	fnd,pr	==>	acc
5	acc,pr	==>	fnd
6	pr	==>	acc
7	pr	==>	fnd
8	fnd	==>	acc,pr
9	acc	==>	fnd,pr
10	fnd,acc	==>	pr

Fig 10: Association Rules stored in database

After obtaining the Role based agent sets it can be associated with other Role based Agents for making the knowledge. Fig 10 shows the Association Rules stored in the database. The best Rules are generated based on the given input transaction dataset values, minimum support and minimum confidence threshold.

Fig. 11 shows the Association Rules with support and confidence. It uses the minimum support value is 0.85 and minimum confidence 0.80 .The best 10 rules are displayed for the confidence values between confidence 1 and 0.8. The rule from 1 to 7 shows the confidence value is 1.0 and the rest from 8,9,10 shows the role based agents transaction with confidence 0.8 respectively.

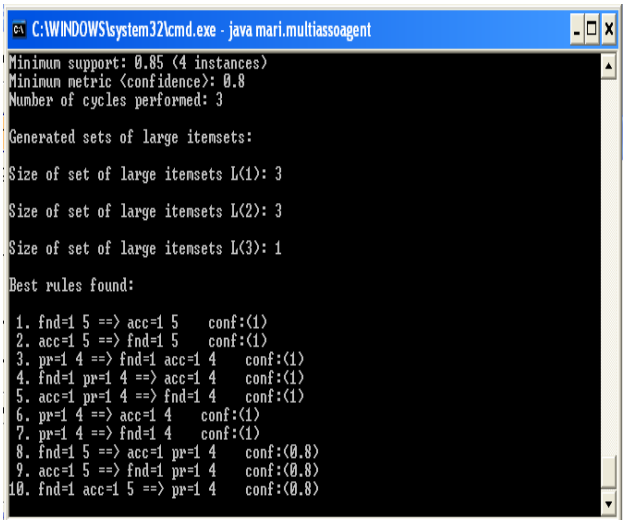

Fig 11: Association Rules with Support and Confidence

6. CONCLUSION AND FUTURE WORKS

This work proposed the system architecture for Distributed Role Based Multi-Agent System. This system is implemented with Banking Domain application and accepts the user request, identifies the role and Agent Facilitator creates the role base mobile agent, transmission of secured agent data and adopts the customer needs to provide the desired results.

The Boolean Association rules are implemented to generate knowledge from frequent Role Based Multi-Agent System. It can be enhanced to implement Quantitative Association Rule to associate role based multi-agents to get more dependency of Distributed role based Agents.

7. REFERENCES

[1] Aurora Vizcanaino,Juan Pablo Soto, Javier Portillo and Mario Piattini. 2007. A Multi-Agent Model to Develop Knowledge Management Systems. In Proceedings of the 40th Hawaii International Conference on System Sciences.

[2] Srinivasan Iyer and Dr.Bhavani Thuraisingam. 2007. Design and Simulation of Trust Management Techniques for a Coalition Data Sharing Environment. In Proceedings of the 11th IEEE International workshop of Future Trends of Distributed Computing Systems (FTDCS'07).

[3] Li Liw,Kevin P.Logan,David A.Cartes and Sanjeev K.. July 2007. Fault Detection, Diagnostics, and Prognostics: Software Agent. IEEE Transaction on Vehicular Technology, Vol. 56, No. 4, pp. 1613-1622.

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

[5] H.H. Bui, S. Venkatesh and D. Kieronska. 2000. A Framework for coordination and learning among team members. In Proceedings of the Third Australian Workshop on Distributed AI (DAI- 97), Perth, Australia.

[6] M.V. Wie. 2000. A probabilistic method for team plan formation without communication. Proceedings of the Fourth International Conference on Autonomous Agents. Barcelona, Spain.

[7] Fabio Belliemine, Giovanni Caire and Dominic Greenwood. 2007. Developing Multi-Agent Systems with JADE. John Wiley & Sons.

[8] Monique Calisti, Daniel Deluca and Andrew Ladd. 2005. An Agent – Based Frame work for Financial Transactions.