# 동적 그룹 바인딩 기반의 모바일 에이전트를 이용한 인텔리전트 분산 플랫폼☆

# Intelligent Distributed Platform using Mobile Agent based on Dynamic Group Binding

마테오 로미오* 이 재 완**
Romeo Mark A. Mateo Jae-wan Lee

## 요 약

오늘날 정보 기술 및 지능형 시스템에서는 분산 데이터베이스로부터 패턴들을 찾고 규칙들을 추출하기 위해 데이터 마이닝 기술을 사용한다. 분산환경에서 데이터 마이닝 기술을 이용해 추출된 규칙들은 동적인 중복, 적응형 부하 균형 및 기타 기술들에서 활용될 수 있다. 그러나 대량의 데이터 전송은 에러를 야기하며 신뢰할 수 없는 결과를 초래할 수 있다. 이 논문은 이동 에이전트를 사용하여 동적 그룹 바인딩을 기반으로 한 인텔리전트 분산 플랫폼을 제안한다. 그룹서비스를 통해 효율적인 객체 검색을 위한 분류 알고리즘을 구현한다. 지능형 모델은 동적 중복을 위해 추출된 규칙을 사용한다. 데이터 마이닝 에이전트와 데이터 압축 에이전트는 각각 서비스 노드 데이터베이스로부터 규칙을 추출하여 데이터를 압축한다. 제안한 알고리즘은 데이터를 전송하기 전에 neuro-fuzzy 분류기를 사용하여 빈도가 적은 데이터 셀을 합하는 전처리 과정을 수행한다. 객체그룹 분류, 서비스 노드 데이터베이스 마이닝, 데이터 압축 및 규칙 추출에 대한 시뮬레이션을 수행했다. 효율적인 데이터 압축 및 신뢰성 있는 규칙 추출에 대한 실험 결과 제안한 알고리즘이 다른 방법들과 비교해 이러한 관점에서 성능이 우수함을 나타내었다.

## Abstract

The current trends in information technology and intelligent systems use data mining techniques to discover patterns and extract rules from distributed databases. In distributed environment, the extracted rules from data mining techniques can be used in dynamic replications, adaptive load balancing and other schemes. However, transmission of large data through the system can cause errors and unreliable results. This paper proposes the intelligent distributed platform based on dynamic group binding using mobile agents which addresses the use of intelligence in distributed environment. The proposed grouping service implements classification method for efficient search of objects. In this paper, the intelligent model uses the extracted rules for dynamic replication scheme of objects. Data compressor agent and data miner agent extracts rules and compresses data, respectively, from the service node databases. The proposed algorithm performs preprocessing where it merges the less frequent dataset using neuro-fuzzy classifier before sending the data. Object group classification, data mining the service node database, data compression method, and rule extraction were simulated. Result of experiments in efficient data compression and reliable rule extraction shows that the proposed algorithm has better performance compared to other methods.

☞ Keyword : Intelligent System, Distributed Object Groups, Mobile Agents, Neuro-fuzzy Systems

## 1. Introduction

Intelligent system models are mostly used in solving complex problem which classical methods cannot achieve. Various researches and industries use software agents to implement intelligent system

* 준 회 원 : 군산대학교 전자정보공학부 박사과정
rmmateo@kunsan.ac.kr
** 종신회원 : 군산대학교 전자정보공학부 교수
jwlee@kunsan.ac.kr
[2007/03/26 투고 - 2007/03/28 심사 - 2007/05/03 완료]
☆ This work is supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD) (KRF-2006-521-D00372).

[1,2]. Agents are provided with intelligence based on the acquired information. The new discovered rules from the environment are used by the agent to change or evolve its task to solve problems. Multi-agent and mobile agent technologies provide developers a new paradigm of designing and implementing software applications. Current researches in agent technology focus on providing industries with a new approach of solving problems in distributed manner, new software tools and automated features from the system [3,4]. In multi-agent models, communications, resource sharing and cooperation on solving a problem are considered [5,6]. Many organizations and researchers are working for the standards, like FIPA [7] and other technologies of multi-agents. However, there are still more challenges in using agent technology. Intelligent models like multi-agents need massive data for inputs in able to acquire the accurate information where transmission of large data through the system can cause errors and unreliable result.

In distributed environment, object group models are designed to manage the system by grouping appropriate objects. The object groups perform cooperation for an efficient service. Communications of object groups reflect the inter-dependence and take place from one group to another. An object group is a set of objects related logically [8]. A group acts as a logical addressable entity where an entity that requests a service from a group is a client of the group. The properties of the object groups are shown below. The effective management of object group is critical for coordination of objects. There are researches integrating their methods in the object group model [9,10]. However, the object group model is lack of group bindings where it enables a fast search of objects by knowing the objects information of each groups.

This work proposes the intelligent distributed platform based on dynamic group binding (DGB). The intelligent model uses the extracted rules from the databases for dynamic replication to provide QoS. There are two agents focused by this research: data miner and data compressor agents. The data miner agent, which is a mobile agent, performs data mining on the service node databases. Data mining technique is used for rule extraction and integrates the rules in the intelligent model. To support data mining, a data compressor agent (DCA) based on neuro-fuzzy classifier is presented. Data miner agent sends request to a local agent manager (LAM) for data. DCA merges the less frequent dataset using neuro-fuzzy classifier. After transmission from DCA, data miner agent decompresses the data and processes data mining. Performance result shows that proposed algorithm is more accurate compared to other methods.

# 2. Related Works

## 2.1. Data Mining using Mobile Agents

Intelligent systems utilize large data across the network of computers to be process and extract knowledge. Data mining in mobile environment using the location-aware agent [11] is used to gather location information from location-based services (LBS). This is done by sending a mobile agent to the LBS from the user agent then the mobile agent performs the classification mining in the database. The result is sent back to the user agent to provide the location information. A location-based service based on multi-agent is presented in [12]. Multi-agents are used for efficiency of location management by using a nearest neighbor search on the hierarchy of the base station. The location agent manager manages the services of the location-based service and communicates to location agents which predict the location of the mobile user. Association rule mining is used to retrieve the previous movements of mobile users and produce a prediction model for locating a mobile user. A collaborative framework is proposed for efficient data mining of location information in the location based-services [13]. The researches mentioned deals with large data and should consider a data compression technique to prevent errors in transmission.

## 2.2. Data Compression Techniques

Data compression is a method of reducing the packets or bits of information and reduces errors of data. There are two types of compression mostly used by researches. Lossless technique [14] is a type of data compression which converts the data into set of binary codes to reduce storage size while lossy compression [15] performs a reduction or transform data to remove noisy and unnecessary results. Huffman coding, a type of lossless compression, uses shorter bit pattern for more common characters and longer bit patterns for less common characters. Wavelet transforms [16] are used for data mining and is one type of the lossy technique. Wavelets are effective to remove noisy data caused by errors. The principal component analysis (PCA) is used by Gerardo [17] in data mining to performed data reduction from data. These two popular methods, loosy and lossless compression, are used on data mining to provide relevant information from the data. However, the disadvantage of using this is time constraint on large data which can affect data transmission.

Fuzzy classification is based on the concept of fuzzy sets, which was conceived by Lotfi Zadeh [18]. Typical fuzzy data analysis discovers rules in large set of data and these rules can be used to describe the dependencies within the data and to classify a new data [19]. Neuro-fuzzy systems are fuzzy classifiers and uses neural networks for learning by performing induction of the structure and adaptation of the connection weights [20,21]. The neuro-fuzzy classification (NEFCLASS) is consisted of 3 layered perceptron. The 1st layer is for inputs ($U_1 = \{x_1,\cdots, x_n\}$), 2nd layer is for generating rules ($U_2 = \{R_1,\cdots,R_k\}$), and 3rd layer is output layer ($U_3 = \{c_1,\cdots,c_m\}$). The values from the input and rule layer are evaluated in the connection of the hidden and output layer. For all output units, the net input $net_c$ is calculated Equation 1.

$$net_c = \frac{\sum_{R\in U_2} W(R,c)\cdot o_R}{\sum_{R\in U_2} W(R,c)} \tag{1}$$

The algorithm also is suitable to estimate the class of an attribute because it deals with a partial membership and process the classifier in its fuzzy system. In our work, we use the neuro-fuzzy algorithm to identify the less frequent data and merge it to the classified dataset. A threshold value is used to adjustment the data compression ratio.

# 3. Intelligent Distributed Platform based on DGB

Intelligent system models are mostly researched systems where approaches are solutions to the previous problems experienced in classical systems. Globe project considers a wide distributed environment where it proposed the support for very large implementation of distributed systems [22].

However, this lacks of intelligence and reconfigurable knowledge. In this study, an intelligent distributed platform based on dynamic group binding is proposed which addresses the necessity of an intelligent approach in distributed environment. Also, the efficiency of data acquisition and rule extraction is considered. The global view of the proposed intelligent model is shown in Figure 1 which consists of three layers physical, logical and application layers.

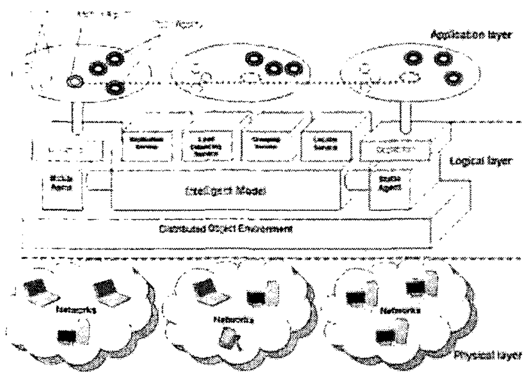



Fig. 1. Architecture of the intelligent distributed platform

The proposed system uses multi-agents to implement the cooperation and automation of tasks and to have transparent behavior of services in managing clients and servers. The physical layer represents networks of different computers like PC, laptop and PDA. The logical layer acts as the middleware where services are transparently executing by the request of clients to the objects. Interaction of clients and object services are handled by the logic layer. The physical layer and application layer does not need to know the configuration on how to find, where to find and how to manage the resources but transparently executes the services to work with the client and servers. Also, the intelligent algorithms are applied inside of this layer. The application layer is consisted of objects and agents utilizing the distributed environment. The proposed intelligent model meets the following design requirements:

- Dynamic replication with load balancing and fault tolerance: quality of service (QoS) is important to clients where object replication is implemented to handle the large number of client requests. In contrast of QoS, the management of the replicated objects is necessary and each load from servers must be balanced in able for the system to provide reliable services and have an optimal performance. Also, fault tolerance schemes use object replicas in case an object service fails to process.
- Scalability and transparency: scalability is important where the expansion of services and additional algorithms are considered. The system is adaptive to the current changes of services and objects. These changes are transparent so to clients and servers do not need to know how to configure and where to find the resources.
- Group management support: object group method offers efficient management of objects. The grouping scheme provides communication to objects within the group and these enhances the search for the appropriate object. In our work, object group uses dynamic binding method through other object groups in providing an efficient search of objects.
- Multi-agent support: multi-agents are used for the intelligent model. An agent acts as an individual which promotes intelligence on how to utilize the objects autonomously. Also, the inter -communication of agents is considered in able to cooperate on task and share the knowledge to other agent.

- Mobility support: mobile agent migration support is necessary for searching of information or providing service to the local node. Also, security is considered for possible alteration data and malicious attacks.
- Data mining support: support for data mining is considered to extract the rules from distributed databases. These rules are used to implement the intelligent system by integrating significant rules to the intelligent model.

## 3.1. Components of the Intelligent Model

### 3.1.1. Replication Service

The replication service (RS) creates and manages the replica objects in the server. RSs within the server are coordinating to other RS managing the replicated objects. If an object has changed its values then RS of that object communicates through other RS informing to change the values of same objects depends on the scheme used. Also, the coordination of the RS decides which server is allowed to create additional objects based on scheme of the intelligent model.

### 3.1.2. Grouping Service

Grouping service (GS) manages the object groups of the proposed system. All objects registered to the GS when initialized. A dynamic classification is done to determine the membership of the object based on its properties. The properties are configured by the application users. This method is proposed to dynamically classify and bind to object groups. Object groups are bounded to other object groups to make the service more efficient on searching the objects and fault tolerant by using object replicas. The reliability of group communication promotes fault tolerance in case of disconnection from an object replica. GS automatically forwards the current request to the disconnected object to another object replica. Figure 2 presents the dynamic group bindings of object groups where group information is shared.

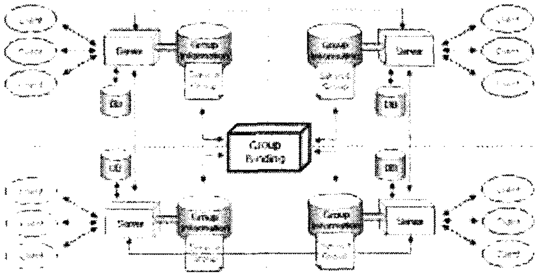


Fig. 2. Dynamic group bindings (DGB) of object groups

### 3.1.3. Load Balancing Service

The load balancing service is responsible for load distribution. A load is defined as a single access of client to an object. In accessing objects, the loads are distributed to the object replicas and follow the threshold values from [10]. An adaptive scheme is used to distribute the loads in able to overcome the problem from two common algorithms of load balancing. The load balancing service coordinates with the sub-components of the group which determines the object replicas in distributing the loads.

### 3.1.4. Locator Service

A locator service is proposed to classify the request and searches the appropriate object. Locator service processes the content of the request in a classification method. Every group has a weight value and the classifier processes the aggregation method. The weight that has the smallest value is the chosen object group to process the request contents by comparing the object's attributes from the group based on Equation 2. c is the summation of the absolute value from the Euclidian distance of the object service ($O_n$) and request content ($RC_n$).

$$c = \sum_{l=1}^{n} \| O_n - RC_n \| \quad (2)$$

### 3.1.5. Data Miner Agent

The main function of data miner agent is to extract rules from database of the service nodes. Before deploying, the data miner agent request verification from the local agent manager to process verification. After sending the data from the data compressor agent, the data miner agent decodes the message and converts it into relevant data. Data mining algorithm is process to generate rules from database of server nodes.

### 3.1.6. Data Compressor Agent

Data compressor agent (DCA) has the function of data preprocessing and compression. LAM request DCA for sending the data to mobile user. First, DCA prepares the data for transmission by compressing the data. The preprocessing of data uses neuro-fuzzy classifier to merge the less frequent datasets. After the preprocessing procedure, Huffman coding is applied to produce shorter bits.

## 4. Data Mining Model using Mobile Agents

The data compressor agent (DCA) performs the lossy and lossless data compression. In the study of Holtz [23], these methods are used on image and video compressions for network transmission. In our study, these methods are used for data transmission. The lossy compression is based on neuro-fuzzy classifier which determines the less frequent dataset and merges it to the more frequent dataset. After the process, lossless technique using Huffman algorithm is applied and sends to the data miner agent. In Figure 3, DCA processes the proposed neuro-fuzzy data compression and data encoding. The compressed data is sent to user agent. Data miner agent decodes the data and performs the Apriori algorithm for the rule extraction.

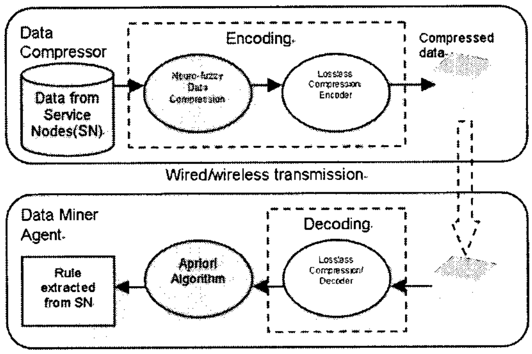



Fig. 3. Procedure of the proposed data compression

### 4.1. Data Compression based on Neuro-fuzzy Algorithm

The proposed data compression classifies less frequent dataset (LFD) and the merges to the appropriate dataset. Numeric values of data are transformed to categorical values before the frequency count. The ($D_x$) is defined as a single combination of values from all data. The *freqCount($D_x$)* counts the frequency of set $D_x$ containing in the transactions of the database. To determine if $D_x$ is an LFD, we get the frequency of the dataset (*FD*) by the frequency count of $D_x$ is divided by the total number of dataset. *totaldata* is the total number of data. The calculation of *FD* is shown in Equation 3.

$$FD = \frac{freqCount(D_x)}{totaldata} \tag{3}$$

A threshold represented by *threshold* refers to the percent value which is set manually. If the quotient of *freqCount*($D_x$) and *totaldata* is less than the threshold then $D_x$ is marked as LFD. The condition of marking the $D_x$ as LFD is shown in Equation 4.

$$LFD(D_x) = D_x \text{ is } LFD \text{ if } FD > threshold \tag{4}$$

LFD is processed for merging after Equation 3. Neuro-fuzzy classification is used to determine which dataset ($D_f$) the LFD will merge. This procedure uses the numeric values to process in fuzzy sets. All dataset that are not LFD become the rule nodes. Fuzzy sets from the linked connection are trained by the dataset which is classified by the rule node. The delta value is determined by Equation 5 to adjust the fuzzy sets

$$\delta_R = o_R(1 - o_R)\sum_{c \in U_3} W(R,c)\delta c \tag{5}$$

After the training of fuzzy sets, the LFD are processed in the structure of neuro-fuzzy. LFD becomes input pattern to calculate the membership function from each rule nodes. The conjunction function of two values in Equation 6 is used.

$$\mu_{A \wedge B}(x) = \min\{\mu_A(x), \mu_B(x)\} \tag{6}$$

The membership values are mapped by the shared links or weights of fuzzy sets. These values are aggregated to the rule nodes. Then the rule node values are compared to each rule node and the one has the greatest value is the chosen node for merging the LFD. Equation 7 shows the procedure of comparing and merging process.

$$D_f = \sum_{i=1}^{n} compare(R_i, R_f),\ if\ R_i > R_f\ then\ f = i$$
$$Merge(D_f, D_x)\ where\ D_x\ is\ LFD \tag{7}$$

```
Node(Rk) = Rule aggregated value
Ck = choosen dataset
for each input xj do
for each μ(i) do
begin
μ(i)ji = μ(i) (xj);
end
for each Rk do
for each antecedent Aj do
begin
if Rk (μ(i)ji) is not 0
then Node(Rk) = current value of Rk + μ(i)ji
end
for each Rkdo
  begin
if Node(Rk) has greatest value
then choose K index
end
return Ck
```

Fig. 4. Neuro-fuzzy classification algorithm

$D_x$ is merged to $D_f$ which is the dataset has the highest membership value. The merging process also implies addition to the frequency count of $D_f$ and removal of $D_x$. Figure 4 shows the pseudo code of the neuro-fuzzy classification choosing the dataset.

## 4.2. Apriori Algorithm

Data mining is applied to extract the rules from databases which are integrated in the proposed intelligent model. Apriori is used to provide a comprehensive association rule extraction from the data. The steps of Apriori algorithm are presented below.

1. Join step - find $L_k$, a set of candidate $k$-itemsets by joining $L_{k-1}$ with itself.
2. Prune step - $C_k$ is generated as superset of $L_k$,

that is, its members may or may not be frequent, but all of the frequent $k$-itemsets are included in $C_k$.

The Apriori property implies that any $(k-1)$-item that is not frequent cannot be a subset of a frequent $k$-itemset; hence, the candidate can be removed. Rules extracted from data mining method are stored in the data miner agent until it returns to original host. After returning to the original host, it sends the rules to be analyzed and processes for reconfiguring the settings of the intelligent model. Figure 5 shows the procedure of rule acquisition from databases and integration of rules to the intelligent model.

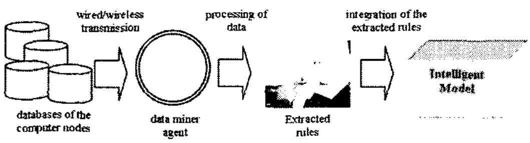



Fig. 5. Data mining process in the distributed databases using mobile agents

# 5. Experimental Evaluation

The data compressor agent and data miner agent were developed in Java programming language. Borland Visibroker was used to implement the intelligent distributed platform in CORBA. The proposed neuro-fuzzy data compression was coded in the data compressor agent while the Apriori algorithm was coded in the data miner agent. Different OS were used like Solaris, Linux and Windows for the heterogeneous resources of databases. Figure 6 shows the initialization of the grouping service and registration of objects. The membership of each objects are registered and each are assigned to a group based on its properties. Also, replication and load balancing services are initialized at this point. Monitoring of objects loads is also presented in Figure 6 where each access of clients is recorded as a load. Data miner and data compression agents were simulated and the results are described in Section 5.2. The performance measures are discussed on the following subsections

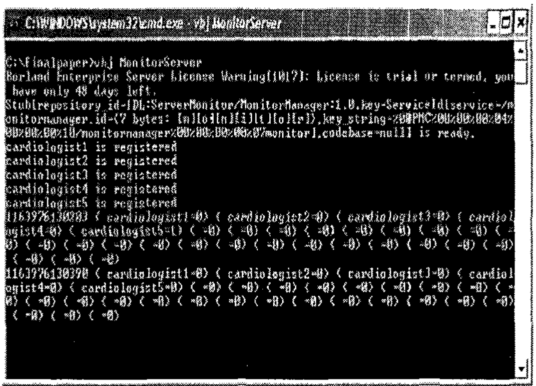



Fig. 6. Screenshot of dynamic group binding by the grouping service

## 5.1. Performance Measures

Server/client operation based on intelligent distributed platform is simulated and the results of client requests are stored in the database. The data were frequency usage of objects associated with the type of service offered from the physical server or service node. The inputs were scaled into least, medium and most requested. There were 1000 datasets gathered to perform the simulation. The following formulations were used to measure the efficiency and reliability of the proposed algorithm.

Let $K$ be a coding of an information source. That is, for each source symbol $a_i$, we have a code word $K(a_i)$and we know the probability $P(a_i)$ of $a_i$. Denoting by $d_i$ the length of the word $K(a_i)$, we can compute the average length $L$of code word:

$$L = \sum_{i=1}^{n} d_i P(a_i) \tag{8}$$

The smallest average length from Equation 8 indicates that it is more efficient. We can say that the proposed data compression provides smaller $L$ of the information because we remove the dataset that has less probability and also merging the LFD from highest probability sets. Processing the new values on Huffman algorithm minimizes the structure of the tree and the word length is reduced. Equation 9 derives the compressed size by dividing the total number of data into the ratio of the output.

$$total\ size = \frac{n}{Huffman \sum_{i=1}^{f} (D_f)} \tag{9}$$

The Apriori algorithm is used to generate association rules. Applying the proposed algorithm removes the less frequent dataset which are considered not important and merged to other dataset. The effects of setting a high threshold value are a relatively high support for the confidence of a datasets because of merging process which implies more association rules are generated. However, some interesting rules might be removed. It is important that the threshold is set appropriately

$$AvgP = \frac{\sum_{i=1}^{n} P_n}{n}, \quad AvgR = \frac{\sum_{i=1}^{n} R_n}{n} \tag{10}$$

In classifier algorithm, recall and precision are performed by cross-validation of the classified instances. To evaluate the accuracy performance of the neuro-fuzzy algorithm, these measurements were used. This is done by calculating the average precisions in Equation 10 where *AvgP* is the summation of precision (*Pn*) of classes divided by the number of classes and average of recall indicated by *AvgR* is the summation of recall (*Rn*) of classes divided by the number of classes. The number of correctly classified instances is used to determine accuracy.

## 5.2. Resultof Data Compression and Rule Extraction

### 5.2.1. Data Compression

Data compression supports mobile agent in data mining the distributed environment. The result presents the efficiency of using the proposed algorithm in data compression method. Figure 7 presents the compression ratio results of using neuro-fuzzy based data compression (NFDC) with threshold values of 1%, 2%, 3%, 4% and 5% and normal compression (NC) without merging. The threshold from the proposed algorithm indicates the compression factor where setting a higher threshold value means a result of smaller data size which is a less constraint on sending the data. Also, the values from the threshold affect the reliability of rule extraction. After executing the neuro-fuzzy preprocessing, a Huffman compression was done. The graph shows that NFDC has a decrease its ratio as the data increases compared to NC. Setting up a higher threshold also implies a high compression rate because the LFD is deducted and the highest probability is incremented by the merge function. However, it is important to set a correct threshold to acquire interestingness of data as well as of acquiring the smaller size data.

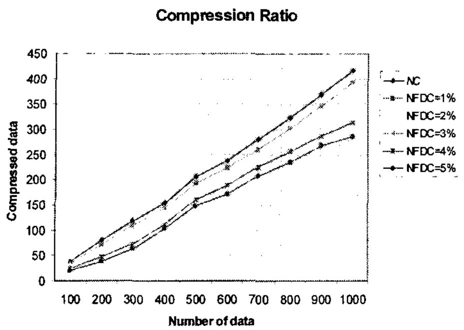



Fig. 7. Result of data compression

The accuracy of classifying the data by the merge function of NFDC was compared to other high accurate classifiers. LFD were classified by using NFDC and other classification algorithms. Results of precision and recall (Equation 10) are presented in Table 2. The result shows that the accuracy of classification of NF is much similar to other highly accurate methods like the MLP. Neuro-fuzzy has the highest output from precision and recall which has an average of 0.943 and 0.936, respectively, compared to Simple Logistic (0.94 (1), 0.925 (3)), MLP (0.937 (2), 0.94 (1)), and FuzzyR (0.67 (4), 0.65 (4)). Moreover, compared to FuzzyR, a fuzzy rule classifier, NF is 28% better accuracy of classifying. Also, performance on processing time of classifying is observed where the NF is the second fastest processing time which has 1.2 seconds compared to FuzzyR (0.8), MLP (2.56) and SL (11.45).

Table 1. Result of precision and recall

| Algorithm | Average Precision | Average Recall |
|---|---|---|
| Neuro-fuzzy(NF) | 0.943 | 0.963 |
| Simple Logistic (SL) | 0.94 | 0.925 |
| Fuzzy Rule (FuzzyR) | 0.67 | 0. 65 |
| Multi-layered Perceptron (MLP) | 0.937 | 0.94 |

### 5.2.2. Rule Extraction

Rules extracted from mobile data mining are integrated in the intelligent model for optimal system performance. After processing the data compression, the data are sent to data miner agent. The result of generated rules using NFDC and normal data was compared. The threshold was set to 1%. The data attributes has a service type indicated by *SERVICE* and object type indicated by *OBJ*x. Table 2 shows the lists of first 5 results from the process. The first rule from Table 2 means that if OBJ2 is least requested then it will imply OBJ3 is most requested with a probability of 99 percent for PC and 97 percent for normal. The same way of explanation can be done to other rules. NFDC generated a total of 38 association rules while 37 for a normal. It is also noted that there are strong similarity of rules obtained between two methods. But observing each confidence, NFDC has an increase of confidence. This is because the data merges on some high frequent dataset and the distribution of frequency of LFD to the datasets. The proposed algorithm retains the original data and provides higher compression rate.

Table 2. Association rules generated from the data of service nodes

| Association rules, showing only 5 | Confidence | |
|---|---|---|
| | NFDC | Normal |
| OBJ2=least req. OBJ3=most req. | 0.99 | 0.97 |
| SERVICE=Type B OBJC3=most req. | 0.99 | 0.98 |
| OBJ1=medium req. OBJC3=most req. | 0.99 | 0.97 |
| OBJ1=most req. OBJ3=most req. OBJ2=least req. | 0.99 | 0.96 |
| OBJ1=most req. OBJ3=least req. OBJ2=most req. | 0.99 | 0.99 |

It is assumed that different content of data also varies in setting the appropriate threshold value. The result from 1% is enough to obtain the reliability of the data for data mining method. Setting up a higher threshold will remove the details from the data and the rules can be unreliable. The rules shown in Table 2 are integrated in the intelligent system. The system analyzes these rules and then the configuration from the services adapts the changes based on the pattern described by the rules. In Table 2, the cases where the object is most requested are determined and assumes that it needs more replication of that to provide QoS which is the responsible of the RM service.

## 6. Conclusion and Future Work

Intelligent system needs large amount of data for accurate analysis. Acquiring large data is a constraint in data mining the distributed system. Data compression is a solution to reduce the constraints in wired or wireless transmission. This work proposes the intelligent distributed platform based on dynamic group binding and introduces a data mining model using mobile agents. The requirements of the intelligent distributed platform and components were discussed. The prototype implementation of the intelligent distributed platform based on dynamic group binding in CORBA is presented. Performance of the system focuses on efficient data acquisition and rule extraction by using mobile agents for the intelligent model. The result shows that the data compression supports data mining in mobile environment by data reduction. Moreover, the rule extracted based on the result of compression is mostly similar to the normal association rule mining which shows the reliability of the rules. These rules are integrated in the intelligent model to reconfigure the proposed system.

This research limits the technical details of rule integration used in the intelligent model and more details from it will be the future research work.

## References

[1] Hendler, J., "Introduction to the Special Issue: AI, Agents, and the Web", Intelligent Systems, IEEE vol.21, No.1, pp.11 - 11, 2006.

[2] Wooldridge, M., Jennings, M., "Agent Theories, Architectures, and Languages: A Survey, In Michael J. Wooldridge and Nicholas R. Jennings", Intelligent Agents, Springer-Verlag, pp. 1-39, 1995.

[3] Marik, V., McFarlane, D., "Industrial Adoption of Agent-based Technologies", Intelligent Systems, IEEE vol.20, No.1, pp.27 - 35, 2005.

[4] Pechoucek, M., Thompson, S.G., Baxter, J. W., Horn, G. S., Kok, K., Warmer, C. Kamphuis, R., Maric, V., Vrba, P., Hall, K. H., Maturana, F.P. Dorer, K., Calisti, M., "Agents in Industry: The Best from the AAMAS 2005 Industry Track," Intelligent Systems, IEEE vol.21, No.2, pp. 86 - 95, 2006.

[5] Jennings, N. R., "Controlling Cooperative Problem Solving in Industrial Multi-Agent Systems using Joint Intentions", Artificial Intelligence, vol.75, No.2, pp.195-240, 1995.

[6] Wittig, T., Jennings, N. R., Mamdani, E. H., "ARCHON - A Framework for Intelligent Cooperation", IEE-BCS Journal of Intelligent Systems Engineering - Special Issue on Real-time Intelligent Systems in ESPRIT, vol.3, No.3, 168-179, 1994.

[7] FIPA, The Foundation of Intelligent Physical Agents, available at http://www.fipa.org

[8] Felber, P., Guerraoui, R., "Programming with

Object Groups in CORBA", Concurrency, IEEE, vol.8, No.1, pp.48-58, 2000.

[9] Joo, S. C., Oh, S. K., Shin, C. S., Hwang, J., "CORBA Based Real-Time Object-Group Platform in Distributed Computing Environment", LNCS 2659, Springer-Verlag, 2003.

[10] Mateo, R. M. A., Yoon, I., Lee, J., "Cooperation Model for Object Group using Load Balancing", International Journal of Computer Science and Network Security, vol.6, No.12, pp.138-147, 2006.

[11] Lee, J., Mateo, R. M. A., Gerardo, B. D., Go, S. H., "Location-Aware Agent Using Data Mining for the Distributed Location-Based Services," Part V, LNCS 3984, Springer-Verlag, pp.867-876, 2006.

[12] Mateo, R. M. A., Lee, J., Yang, H., "Optimization of Location Management in the Distributed Location-based Services using Collaborative Agents", Part III, LNCS 3982, Springer-Verlag, pp.17-22, 2006.

[13] Mateo, R. M.A., Lee., M, Lee, J., "Location-aware Data Mining for Mobile Users based on Neuro-fuzzy Systems," LNAI 4332, Springer -Verlag, pp.1269-1278, 2006.

[14] Holtz, K., and Holtz, E., "Lossless data compression techniques", in: Proceedings of WESCON/94, 'Idea/Microelectronics, pp.392-397, 1994.

[15] Berger, T., Gibson, J. D., "Lossy Source Coding", IEEE Transactions on Information Theory, vol.44, Issue 6, pp.2693-2723, 1998.

[16] Li, T., Ma. S., Ogihara, M., "Wavelet Methods in Data Mining", The Data Mining and Knowledge Discovery Handbook, pp.603-626, 2005.

[17] Gerardo, B. D., Lee, J., Ra., I., Byun, S., "Association Rule Discovery in Data Mining by Implementing Principal Component Analysis", LNCS 3397, Springer-Verlag, pp.50-60, 2005.

[18] Zadeh, L. A., "Fuzzy Sets, Information and Control", pp.338-353, 1965.

[19] Kruse, R. Bolgelt, C., Nauck, D., "Fuzzy Data Analysis: Challenges and Perspectives", in: Proceedings of the 8th IEEE International Conference on Fuzzy Systems, IEEE Press, 1999.

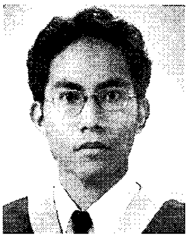
[20] Klose, A., Nürnberger, A., Nauck, D., Kruse, R., "Data Mining with Neuro-Fuzzy Models", Data Mining and Computational Intelligence, Springer- Verlag, pp.1-36, 2001.

[21] Nauck, D., Kruse, R., "NEFCLASS - A Neuro-Fuzzy Approach for the Classification of Data", in: Proceedings of ACM Symposium on Applied Computing, Nashville, 1995.

[22] Homburg, P., Van Steen, M., Tanenbaum, A. S., "An Architecture for a Wide Area Distributed System", in: Proceedings of 7th ACM SIGOPS European Workshop, pp.75-82, 1996.

[23] Holtz, K., "Digital Image and Video Compression for Packet Networks", available at http://www.autosophy.com/icomart.htm.

## ◐ 저 자 소 개 ◑



**마테오 로미오(Romeo Mark A. Mateo)**
2004 West Visayas State University, Philippines BS in Information Technology
2007 Kunsan National University, South Korea Master of Engineering major in Information and Telecommunications
2007~ currently Kunsan National University, South Korea Graduate student in Ph.D course
Research interest : Distributed systems, data mining, fuzzy systems, multi-agents, mobile computing
E-mail : rmmateo@kunsan.ac.kr

**이 재 완(Jae-wan Lee)**
1984년 중앙대학교 이학사-전자계산학
1987년 중앙대학교 이학석사-전자계산학
1992년 중앙대학교 공학박사-컴퓨터공학
1996~ 1997 한국학술진흥재단 전문위원
1992 ~ 현재 군산대학교 전자정보공학부 교수
관심분야 : 분산 시스템, 운영체제, 실시간 시스템, 컴퓨터 네트워크, 데이터마이닝 등
E-mail : jwlee@kunsan.ac.kr