The method of using database technology to process rules of Rule-Based System

BaoWei Zheng, Jeongmo Yeo

***Abstract*—The most important of rule-base system is the knowledge base that determines the power of rule-base system. The important form of this knowledge is how to descript kinds of rules. The Rule-Base System (RBS) has been using in many field that need reflect quickly change of business rules in management system. As far, when develop the Rule-Based System, we must make a rule engine with a general language. There are three disadvantage of in this developed method. First, while there are many data that must be processed in the system, the speed of processing data will become very slow so that we cannot accept it. Second, we cannot change the current system to make it adaptive to changes of business rules as quickly as possible. Third, large data make the rule engine become very complex. Therefore, in this paper, we propose the two important methods of raising efficiency of Rule-Base System. The first method refers to using the Relational database technology to process the rules of the Rule-Base System, the second method refers to a algorithm of according to Quine McCluskey formula compress the rows of rule table. Because the expressive languages of rule are still remaining many problems, we will introduce a new expressive language, which is Rule-Base Data Model short as RBDM in this paper.**

***Index Terms*—Rule-Based System (RBS), Relational Database, Business Rule, Rule Based Data Model (RBDM).**

I. INTRODUCTION

Rule-based system is defined as represent knowledge in terms of a bunch of rules that tell us what we should do or what we could conclude in different situations[2]. A rule-based system consists of a bunch of IF-THEN rules, a bunch of facts, and some interpreter controlling the application of the rules, given the facts. There are two kinds of rules systems: forward chaining systems and backward chaining systems. In a forward chaining system we start with the initial facts, and keep using the rules to draw new conclusions given those facts. In a backward chaining systems we start with some hypothesis (or goal), we are trying to prove, and keeping for rules that would allow us to conclude that hypothesis[3].

Manuscript received November 12, 2009; revised November 25, 2010; accepted December 1, 2009

Zheng BaoWei is with the Department of Information Engineering, Pukyong National University, Busan, 607-737, Korea (Tel: +82-10-9946-8311, Email: buging2007@hotmail.com)

Rule-base systems provide a powerful tool for knowledge specification and development of practical applications. However, although the technology of RBS becomes more and more widely applied in practice, solutions based on the use of rule-based systems are still not well-accepted by some industrial engineers[5]. This is so mainly due to three reasons: first, it sometimes needs complex rule patterns and rule reference mechanism. Second, the correct of them requires much intuition and domain experience, and knowledge acquisition still constitutes a bottleneck for many potential applications. Third, a serious problem concerning RBS is a consequence of the fact that a complete analysis of properties remains still a problem, especially supporting the design stage rather than the final verification. For the third season, we can select kinds of three accepted language, which are propositional logic, attributive logic, and first order logic. Depending on the accepted language, the rule can have different expressive power. A very popular solution is based on the use of various forms of attributive-based languages. This makes the notation similar to the one we used in Relational Database Systems (RDBS).

In this paper, we use the matured data manipulation technology to process the rules that are specified by Rule Base Data Model. And all of the data operation will be completed by database. It is very advantage to use the Rule Based Data Model (RBDM) language for representing rules[1][7]. Any facts of determining a action or conclusion are considered as a variable or a RDBS table that have a relationship with rule entity, and any rule is manipulated as a row of RDBS. Advantage of the expressive language concerns two sides: the first, we can use matured data manipulation technology for rule processing, The second, there is no longer need to design a complex rule engine for processing number of rules.

Through analysis the characteristics of the rule table, we find out there are many repeat data value in every column. In order to reduce the number of rows in the rule table we applied the application of Quine McCluskey algorithm. We can make use of the algorithm for compressing the number of rows of the rule table. In general, we can reduce the number of rows of the rule table to 1/3 of before compressed. The compressed technology not only save the stored space but also advance the efficiency of accessing data.

II. ARCHITECTURE OF RULE-BASED SYSTEM

In this part, we will introduce the architecture of the Rule-Based System. Because according to the characteristics of rule Rule-Based System have been divided into Forward Chaining System and Backward Chaining System. This part including not only the architecture of Forward Chaining System and Backward Chaining System, but also concerning the architecture of we have researched named as Database-Based Rule System.

A. Architecture of Forward Chaining System

In a Forward Chaining System the facts in the system are represented in a Working Memory which is continually updated with add operation and delete operation. Rule in the system represent possible actions to take when specified conditions hold on items in the Working Memory. The conditions are usually patterns that must match items in the Working Memory, while the actions usually involve adding or deleting operations from the Working Memory.

In order to simplify the introduction of the architecture of it, we make a specific example of practical rules in following Fig.1.

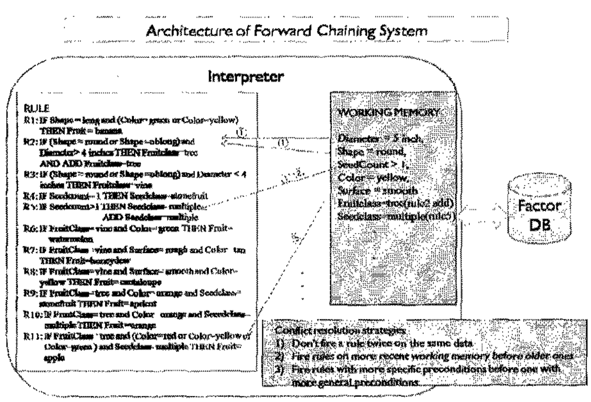


Fig. 1. The architecture of Forward Chaining System.

As the Fig.1 specified, the interpreter controls the application of the rules, given the Working Memory, thus controlling the system's activity. It is based on a cycle of activity sometimes known as a *recognise-act* cycle. The system first checks to find all the rules whose conditions hold, given the current state of working memory. It then selects one and performs the actions in the action part of the rule. The selection of a rule to fire is based on fixed strategies, known as *conflict resolution* strategies, which including contents are being specifying in bottom rectangular of the Fig.1. The actions will result in a new working memory, and the cycle begins again. This cycle will be repeated until either no rules fire, or some specified goal state is satisfied.

B. Architecture of Backward Chaining System

In a Backward Chaining System given a goal state to try and prove the system will fist check to see if the goal matches the initial facts given. If it does, then that goal succeeds. If it doesn't the system will look for rules whose conclusion match the goal. One such rule will be chosen, and the system will then try to prove any facts in the preconditions of match the rule using the same procedure, setting these as new goal to prove.

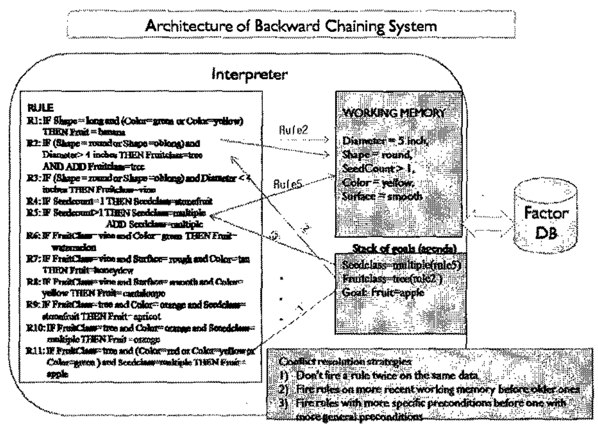


Fig. 2. The architecture of Backward Chaining System.

C. Architecture of Database Rule System

In Database Rule System, we have no more need of developing a rule engine. To contrary, we use the database to instead of rule engine for processing all of rules which must be executed by rule engine before. Each of factors that determine decisions or actions of rule must be considered as table to save into database. And every rule case considered as a row to save into rule table. Every rule case is constructed by combination of many factors.

The database can identify the unique decision or action according to input values of factors. When we use some given condition to access data from database, have no more need of writing much general language rather than writing SQL. Application of the database not only make the executed speed has been advanced but also simplify the complexity of the Rule-Base System. The specific architecture of Database Rule System is specifying as following.

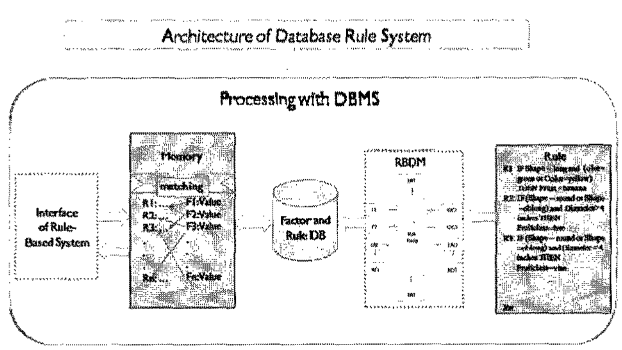


Fig. 3. The architecture of Database Rule System.

In Database Rule System, all of Rule elements can be designed into Rule Base Data Model according to the knowledge of Relational Database. And every entity in Rule Base Data Model can be converted into a physical table that saved into database. User can input some certain factor values through the interface of Rule-Based System. When there are some factor values in Memory, DBMS start checking to find all the rules whose conditions hold, given the current state of memory. It then selects one and performs the actions in the action part of the rule. DBMS as a core part in the Database Rule System, which complete all of processing procedure.

III. RULE BASE DATA MODEL

According to characteristic of rule we have divided the rules into three types, such as Multi Dimension, Rule Matrix, and Rule Tree. In this unit, we will introduce characteristics and adaptive situation of every types of rule in detail.

A. Multi Dimension

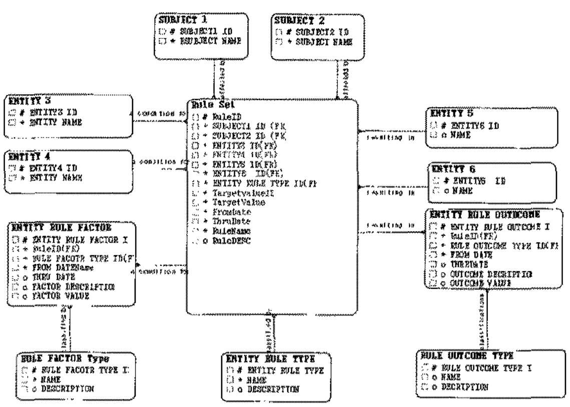


Fig. 4. Multi Dimension Data Model.

We use this mode as followed situation: when an enterprise is interested in capturing how business rules for specific subject areas are integrated to other entities in their data models. There is a need of have a flexible data model structure capturing rules. The enterprise does not have a metadata environment or business engine to capture rules. The enterprise wishes to capture additional business rule factors that are not directly tied to entities that exist in the data model, or when there is a need to accommodate future types of factors that may emerge. As an intermediary solution can be used to instead a business rules engine or metadata repository environment. There are situations where the data professional needs to show a statement of scope to other IT professionals.

The purpose of the data model pattern is to maintain business rules data such as the business rule statement, business rule factors, and business rule outcomes for particular situation, problem area, or business subject area. Many situations such this one occur while data modeling and this pattern allows data professionals to address each situation with a common template to model rules, and their associated factors and outcomes, and integrate these rules into their data model.

B. Rule Matrix

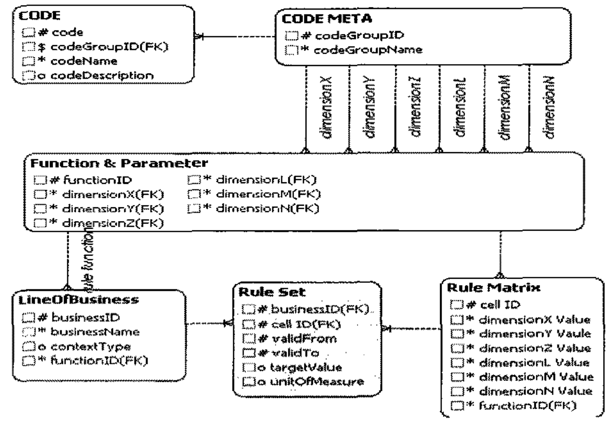


Fig. 5. Rule Matrix Data Model.

In the rule matrix data model, all of factors information that affect the conclusion of rule are grouped into different group and managed by CODE META entity. Information of factor will be managed by CODE entity. Factors that Affect conclusions of rule can be considered as variables which can be accepted from entity CODE META. Function of determining conclusions of rule and it need variables as a row will be managed in Function & Parameter Table. Every specific rule case as a row can be saved in Rule Matrix. It is a table of matching function and variables. In general, there is amount of data in the rule set table of matching line of business and rule case.

It is a very import type of Rule Base Data Model that makes use of Data Model for specifying rules of Rule-Based System. The largest of advantage of the model is what it can be applied in any situations. However, the largest of disadvantage of the mode is what it cannot be considered as an embedded model insert into existed model, rather than as a new data model must be designed when we develop a new Rule-Based System. If we use the data model, we have no need of developing a complex rule engine. All of operations of Rule engine must execute before, we can apply Database technology to instead completely it.

C. Rule Tree

When some certain results of conditions are preconditions of other conditions, propositional logic calculus operation must be executed by special mechanism in this type of Rule-Based System. The special mechanism can be called rule engine, so we must develop a rule engine for complete propositional logic calculus operation. It is the biggest disadvantage of the data model that we make use of it to specify the rules.

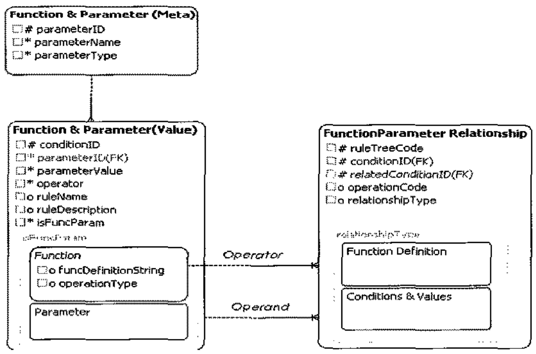


Fig. 6. Rule Tree Data Modeling.

IV. ALGORITHM OF COMPRESSING ROWS OF THE RULE TABLE

In the Rule-Based System, the most difficult problem is how to use an effective language for specifying rules, and how to access data from the rule table as quickly as possible. So we will introduce a new method to compress the rows of rule table, in order to improve executed speed of accessing data that satisfied some certain condition.

A. Rule table with atomic values of attributes

The form of rules set as following:

$$ri: (A1 = di1) \wedge (A2 = di2) \wedge \ldots (An = din) \rightarrow \rightarrow H1 = hi1 \wedge H2 = hi2 \wedge \ldots Hm = him. \quad (1)$$

Taking into account the advantage of the uniform form of all the rules in the system, the set of rules can be specified in a transparent, tabular form. The definition of rules as showed in table1. We have represented k uniformly structured rules in the rule table.

TABLE I
RULE TABLE WITH ATOMIC VALUE OF ATTRIBUTIVE

r	A_1	A_2	…	A_j	…	A_n	H_1	H_2	…	H_m
r_1	d_{11}	d_{12}		d_{1j}		d_{1n}	h_{11}	h_{12}		h_{1m}
r_2	d_{21}	d_{22}		d_{2j}		d_{2n}	h_{21}	h_{22}		h_{2m}
…	…	…		…		…	…	…		…
r_i	d_{i1}	d_{i2}		d_{il}		d_{in}	h_{i1}	h_{i2}		h_{im}
…	…	…		…		…	…	…		…
r_k	d_{k1}	d_{k2}		d_{kj}		d_{kn}	h_{k1}	h_{k2}		h_{km}

Unfortunately, for a number of more complex, realistic applications, it is not enough to stay at the level of atomic values only. There are at least the following three reasons for that: many attributes are in fact generalized attributes—they can take several values (a subset) from the domain at a certain instant of time, i.e.; Knowledge specification (both in case of rules and facts) with non-atomic values can be far more concise than in the case of atomic values; It is often impossible to specify precise definition of preconditions, and one needs to have the possibility to specify attribute values as belonging to intervals, subset of the domain, etc.

B. Rule table with set values of attributes

Every attribute can take a set of values in rule table. Let $u \in U$ be an object, $Ai \in A$ be an attribute and let $t \subseteq Di$ be some subset of the domain of Ai. Two most typical atomic formulae will be considered, i.e. the one with equation of the form

$$Ai(u) = t$$

and the one with 'belongs to' relation

$$Ai(u) \in t.$$

Both of the formulae are atomic formulae of SAL. On the base of atoms, appropriate selectors can be constructed to form preconditions of the rules.

Consider a set of rules, each of the form

$$ri: (A1 \in ti1) \wedge (A2 \in ti2) \wedge \ldots (An \in tin) \rightarrow H1 = hi1 \wedge H2 = hi2 \wedge \ldots Hm = him. \quad (2)$$

Taking into account the advantage of the uniform form of all the rules in the system, the set of rules can be specified in a transparent, tabular form (one resembling database format). The definition of (a basic form of) a set value of rules can be specified in below table 2.

The table 2 represents k uniformly structured decision rules with non-atomic values of attributes. The matrix notation can be also used as in the case of rule table. In the table, every t represents a set of values that every attributive can take all of possible value.

TABLE II
RULE TABLE WITH SET VALUE OF ATTRIBUTIVE

r	A_1	A_2	…	A_j	…	A_n	H_1	H_2	…	H_m
r_1	t_{11}	t_{12}		t_{1j}		t_{1n}	h_{11}	h_{12}		h_{1m}
r_2	t_{21}	t_{22}		t_{2j}		t_{2n}	h_{21}	h_{22}		h_{2m}
…	…	…		…		…	…	…		…
r_i	t_{i1}	t_{i2}		t_{il}		t_{in}	h_{i1}	h_{i2}		h_{im}
…	…	…		…		…	…	…		…
r_k	t_{k1}	t_{k2}		t_{kj}		t_{kn}	h_{k1}	h_{k2}		h_{km}

The interpretation of set values of attributes is as follows. Let t be a subset of the domain of attribute Ai, $t = \{d_1, d_2, \ldots, d_j\}$. In preconditions, a selector like

$$Ai \in t$$

can be expressed as

$$Ai = d1 \vee Ai = d2 \vee \ldots \vee Ai = dj$$

C. algorithm of compressing rows of the rule table

In this part, we will introduce the important algorithm of compressing row of the rule table. The k-map algorithm has a very big disadvantage that considered as rely on pattern recognition. And Quine McCluskey Method can be only used in Boolean function. We have analyzed the characteristics of the rule table and the two famous algorithms, and then applied the Quine

McCluskey Method for compressing the rows of rule table of rule-based system. There are three 3general steps in our new algorithm: 1) generate prime rule table, and according to the types of conclusion of the rule table group all of rows, 2) generate list of 0-subcubes in every conclusion units, 3)if it possible, generate list of 1-subcubes base on the list of 0-subcubes.

1) Generating prime rule table.

Because the base of the algorithm is function that makes use of giving variables to calculus conclusions, we can consider the unique format of the rulei: $\#p_1 \wedge \#p_2 \wedge \ldots \wedge \#p_n \rightarrow \#h1$ as a function. Every rule case can be considered as a combination of defined variables in the rule table. We can consider every column as a subcube. Thus, the structure of a perfect example rule table as following:

TABLE III
RULES TABLE OF OPTICIAN EXAMPLE

	No	Ag	Sp	As	Tp.r.	Decn
	1	y	m	y	n	H
G1	2	y	n	y	n	H
	3	p	m	y	n	H
	4	q	m	y	n	H
	5	y	m	n	n	S
G2	6	y	n	n	n	S
	7	p	m	n	n	S
	8	p	n	n	n	S
	9	q	n	n	n	S
G3	10	y	m	n	r	N
	11	y	m	y	r	N
	12	y	n	n	r	N
	13	y	n	y	r	N
	14	p	m	n	r	N
	15	p	m	y	r	N
	16	p	n	n	r	N
	17	p	n	y	r	N
G3	18	p	n	y	r	N
	19	q	m	n	r	N
	20	q	m	n	r	N
	21	q	m	y	r	N
	22	q	n	n	r	N
	23	q	n	y	r	N
	24	q	n	y	r	N

In above table 3, Ag denotes age, Sp denotes spectacle, As denotes astigmatic, Tp. r. denotes tear produce rate, desn denotes decision.

According to the value types of decision we will group all rows of the rule table into three groups, 1~4 as G1, 5~9 as G2, 10~24 as G3. If there are many rule cases in the rule table even we don't group them the algorithm can also compress rows, but it will need more time than grouping them. In practical application, not only there are many value types of decision but also there are many row cases in the rule table. So we cannot group all of rows into different group. In order to simplifying application of the compressive technology of the rule table we don't need to divide all rows into different group.

From the first row of one group or from the first row of the rule table (not exist groups), find any 2 rows that differ in exactly one variable, simply by comparing the rows with other rows in one group or in the rule table. For example, row 1 and row 2 in the list of 0-subcubes differ by one variable (the spectacle variable). Therefore, we can compress row 1 and row 2 together into a row in the list of 0-subcubes and the differing variable (the variable spectacle for this case) is replaced with an underscore. The underscore is called a do not care. Using the same technique we can generate the rest of the 0-subcubes for G1, G2 and G3 or all of rows of the rule table.

2) Generating a list of 0-subcubes.

TALBE IV
TABLE OF LIST OF 0-SUBCUBES

	No	Ag	Sp	As	Tp.r.	Decn
	1	y	-	y	n	H
G1	2	-	m	y	n	H
	3	-	m	y	n	H
	4	y	-	n	n	H
	5	-	m	n	n	S
	6	-	n	n	n	S
G2	7	p	n	n	n	S
	8	-	n	n	n	S
	9	y	m	-	r	S
	10	y	-	n	r	N
	11	-	m	n	r	N
	12	-	m	y	r	N
	13	y	-	y	r	N
	14	-	m	y	r	N
	15	-	m	y	r	N
	16	y	n	-	r	N
G3	17	-	n	n	r	N
	18	-	n	n	r	N
	19	-	n	y	r	N
	20	-	n	y	r	N
	21	p	m	-	r	N
	22	p	-	n	r	N
	23	-	m	n	r	N
	24	y	-	y	n	N
	...	...	...	...	...	...

There are many repeat rows in the table, before generate a list of 1-subcubes, the algorithm must delete all repeat rows from the table. Otherwise, it can give bad effect to procedure of compressing rows.

3) Generating a list of 1-subcubes.

Generate a list of 1-subcubes. Again, comparing the rows in every group of 0-subcubes generates these 1-subcubes. And if the rows differ by only one variable they are grouped into the same group. Note that an underscore is not an indication of the value of a variable but is rather a placeholder for a variable that has already been eliminated from the expression.

TABLE V
RESULT OF COMPRESSING

Group	No	Ag	Sp	As	Tp.r.	Decn
G1	1	y	-	y	n	H
	2	-	m	y	n	H
G2	3	y	-	n	n	S
	4	-	n	n	n	S
	5	p	-	n	n	S
G3	6	-	-	-	r	S
	7	p	n	y	-	N
	8	q	m	n	-	N
	9	-q	n	y	-	N

When you see an underscore under the subcube(column) values, this means the variable has been eliminated from the expression and the variable will not be in the a certain row. For example, in the first row of group G1 we have condition age=y and astigmatic=y and tear produce rate=n and condition age=y and spectacle=m and astigmatic=y and tear produce rate=n return a same decision H. therefore, value of the column spectacle can be ignored in the row.

In order to simplify the algorithm of compressing rows of rule table, we have token a simple example. There are four factors which affect the decision of rules in the example. And all of combinations of the factor values are 25 rows in rule table. Then we have applied the algorithm to compress the rows of the rule table, and reduced the rows of the rule table to 9 rows. Through many other experiments we have found out the rows of the rule table can be compressed to 1/3 on average. The amount of space consumed by uncompressed data will be two to three times larger than that of the compressed data. In fact, in many cases performance may improve due to the reduction in I/O since DBMS will have to access fewer blocks. Further, the buffer cache will become more efficient by storing more data without having to add memory.

V. CONCLUSIONS

After that we have analyzed comprehensively the characteristics of the rule, existed three kinds of problems in current Rule-Based System, and the possibility of applying the database technology to process the rules, we have designed a new architecture of Rule-Based System, it is Database Rule System. It is very advantageous to make use of elements of Database technology for simplifying the operation of current Rule-Based System. The combination of Rule-Based System elements and Relational Database technology can produce potential practical importance in extended it to practical application. According to the characteristics of rule we have designed three kinds of Rule Base Data Model with Data Model Language. It can be applied into any situations where we wish to capture data of rules. Taking account to advance the execute speed of accessing data, have found out that there are many repeat values in every column of rule table, the new algorithm of compressing rows of rule table have been researched out in this paper.

When we make use of DBMS to process the rules, how to the table type will affect efficiency of accessing data, how much overhead of data operation we must pay. And when we insert, update, delete data in table, how to the specified processes will be executed in compressed environment and other problems will be researched in next paper.

REFERENCES

[1] En-core consulting. Rule Base Data Model Seminar, 2008. www.en-core.com.

[2] Editor-in-chief and Prof. Janusz Kacprzyk. Logical Functions for Rule-Base Systems, second edition. Spinger. 2006.

[3] Ahmed T. Sadik. Premises Reduction of Rule Based Expert System Using Association Rules Technique. International Journal of Soft Computing 3(3): 195-200, 2008.

[4] Oracle Corp. Oracle® Fusion Middleware User's Guide for Oracle Business Rules 11g Release 1 (11.1.1). 2009.

[5] Alison Cawsey. The architecture of forward chaining Rule-Base System and backward chaining Rule-Base System. Department of Computing and Electrical Engineering Heriot-Watt University Edinburgh EH14 4AS, UK. 2007.

[6] David C.Hay. Data Model Patterns A Metadata Map. Morgan Kaufmann Publishing, 2007.

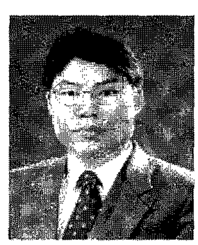
[7] Len Silverston, Paul Agnew. The Data Model Resource Book. Wiley Publishing, Inc. 2009. pp. 411-468.

[8] Steve Hoberman & Associates, LLC. Data Modeling Master Class. 2008. pp. 112-280. www.stevehoberman.com

[9] Stephande Faroult & Peter Robson. The Art of SQL. Publishing House of electronics industry. 2008. pp.167-190.

[10] Lee Huw Sick. New Written, Large Scale Database Solution. Publishing En-core consulting. 2005. pp.1-99, 323-399.

BaoWei Zheng.
Xi'an, China, December 12, 1980.
2004. Department of Computer Engineering, Xidan University, Bachelor. 2007. Department of Computer Multimedia Engineering, Uiduk University, Master. 2007-Now. Department of Information Engineering, Pukyong National University, Doctor. Major Field of Study: Mobile interface, Database, Rule-Based System.



Jeongmo Yeo.
Busan, Korea, May 04, 1956.
1980. Department of Electronics Engineering, Dong-A University, Bachelor. 1982. Department of Electronics Engineering, Pusan National University, Master. 1993. Department of Electronics and Computer Engineering, University of Ulsan, Doctor. 1986-Now. Division of Electronics, Computer, and Telecommunication Engineering, Pukyong National University, Professor. Major Field of Study: Database, Electronic Commerce, EMFG.