Design and Implementation of the EEIS Considering the Load of DB Server

Chang-Geun Kim*, Byeong-Jin Choi**, and Han-Ho Tack***

* Department of Computer Science, Jinju National University
150, Chilamdong, Jinju 660-758, Kyongnam, 660-758, Korea
E-mail : cgkim@jinju.ac.kr
** Department of Academic information computer center
E-mail : bjchoi@jinju.ac.kr
*** Department of Electronic Engineering, Chinju National University
E-mail : fmtack@jinju.ac.kr

Abstract

Current Internet system of the entrance exam information to the university that is depending on searching key to solve the overloaded problem in the network and DB server or other tools to support HTML edit, haven't satisfied user's wants by supplying uniformed searching system. So this thesis will establish EEIS(Entrance Examination Information System) to prevent database overload phenomena when many users request a great amount of data at the same time and improve the decrease of speed and overload problem in DB server. EEIS play a role of bridge between outside client and DB server by placing VVS(Virtual View Server) between web server and DB server. By that method this system give users several usefulness in convenience and variety by supplying realtime data searching function to user. EEIS also give inner system manager more efficiency and speed in control the management system by solving those problem. This system is design and implementation to satisfy user's desire and give them more convenience and bring up the confidence of university that adopt this system at the end.

Key words : Overload, DB Server, Searching, VVS(Virtual View Server)

1. Introduction

Nowadays, it is expected to provide the information system of which can provide the information anywhere anytime thanks to the wide availability of internet service. However, the current generation of the remote support system for database search, of which need to have the cooperation between several data bases over internet, only provides the unique key search system in order to prevent the overload in its database and/or network infrastructure, or only provides HTML editing service, and it does not fully support the user demands[5].

In this paper, the authors have implemented VVS (Virtual View Server) between Web server and D/B server in order to correct the above problem, and VVS is also an important part of EEIS (Entrance Examination Information System). It is not necessary to have the independent system for VVS as long as the web server in question does have the adequate web server and the enough hard disk capacity.

The primary procedure of VVS is to deploy the required program such as DB server and the associated DB engine and then to implement the view table from the required data for CGI in order to prevent from instability and overloading of DB server. The network speed and efficiency of EEIS system have been improved for internal users. Also this system provides various realtime information for the external users. As a result of implementing this system, the university wide systems' capabilities have been proven as reliable and secure. At the same time it improves the level of the user satisfaction[8][10].

2. Related work and motivation of the study

2.1 CGI

CGI(Common Gateway Interface) is the protocol for the exchange of data between web server and other programs. Common gateway can be described as route to exchange between multiple programs of which are running in different network domain. And Interface represent the "Programming Interface" for this type of data route.

Its tasks are achieved by common variables and standard input & output[4][6][7][9][11].

The figure 1 describes the overall flow of data among HTML, CGI, and DB. That is the request for URL by web client(browser) would produce the request header and send it to web server.

When the above request has been passed on to Web server, the request header would be decoded whether it is the request for the specific web page or the request for the forking the

new process for CGI program. If it is the request for the process of CGI program(or scripts) then the web server would fork the new process for its implementation[4].
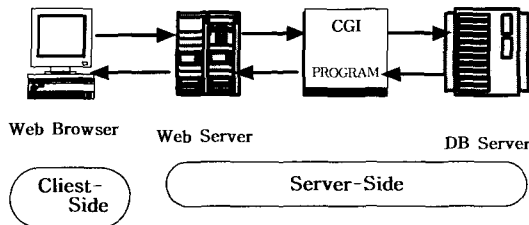

Fig. 1. Interface flow of HTML & CGI & DB

The differences between the CGI program and the general program are as follows.

1. The general program process the request within its own memory space in realtime and ensure the correct serialization. That is its variables would be stored local memory space whether it is heap or stack. However, CGI utilize the request and response architecture (as in asynchronous transfer), and its connection would be disconnected after final acknowledgement.

2. The non CGI programs are normally processed at the clients-side, and CGI programs are associated with Web Server in order to expand the functionality of Web server.

Because of the above feature, there is a need for a protocol between Web server and CGI program. And the protocol defined and used for this purpose is CGI (common gateway interface). As the primary purpose of CGI is to act as an interface between Web server and its clients, there isn't any restriction on the development tools for CGI program development. Therefore any development language (such as C, C++, Perl, Shell script) can be utilized, as long as there is a mechanism for standard input and output capability[4][6][7][9].

2.2 Related of CGI and DB system

Data flow between CGI and DB system can be depicted as figure 2. Since CGI and DB system have been connected, there is no need for user to learn complex command sets. Also it enables user to request the information from DB when he/she needs them through Web interface.

The method for accessing D/B system can be grouped into two methods. The first method is to use API functions provided by D/B system and the second method is to utilize the extra D/B gateway between the web interface and D/B system.

The first method can be described as accessing D/B server by using DB API functions within CGI programs and/or Web server API. For example, it is possible to define and utilize the necessary D/B within C program using ESQL/C for Informix D/B.

For the DB Gateway method, it is possible to embed the backend program which is provided by DB vendors. The "backend" program can be described as "An independent middle ware for easy DB handling from the main DB system process". For example, mSQL provides W-mSQL as a backend program. By using W-mSQL of which acts as an

intermediary, it eliminates the needs for direct request between mSQL and D/B system being developed[4][8][10].
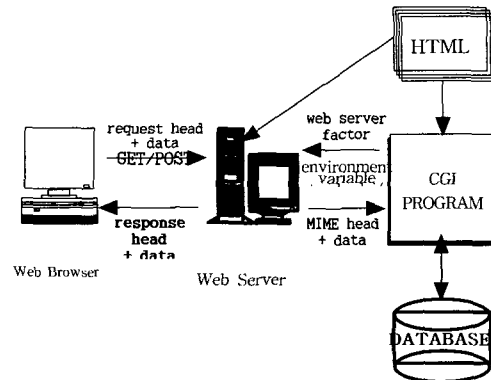

Fig. 2. Related flow of CGI and DB system

Figure 3 depicts the process of embedding mSQL API.

First of all, connect to mSQL DB server through TCP/TP by issuing msqlConnect(). And then it is followed by selecting user DB by msqlSelectDB() . Next, the desired query would be send to DB by msqlQuery(). Once, the enquery process has been finished, the requested values would be passed on to the real memory space from Cach buffer by using msqlStoreResult(). The fetched data row would be displayed on Screen by msqlNu mRows() or msqlFetchRow(). Finally, the allocated memory would be deallocated by msqlFreeResult() and disconnected from DB Server[4][10].
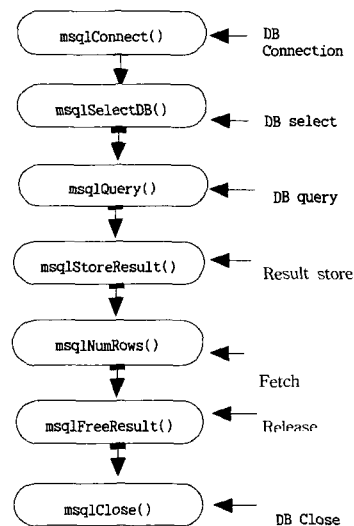

Fig. 3. The flow of mSQL API

2.3 XML

XML(eXtensible Markup Language) is developed by W3C(World Wide Web Consortium) in order to improve the functionalitiesof SGML(Standard Generalized Markup Language) and HTML. It can utilize the merit of Tag and structured documentation from SGML and the one of easy provision of providing hypermedia documents in web from HTML[2][3][11][12].

3. Design and analysis of system

3.1 Environment of system

EEIS system consists of Web server , VVS and DB server, it can be depicted as figure 4.
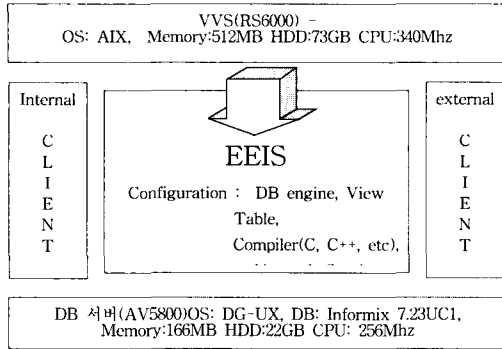


```
VVS(RS6000) -
OS: AIX,  Memory:512MB HDD:73GB CPU:340Mhz
```

```
Internal                          external
   C            EEIS                 C
   L                                 L
   I     Configuration :  DB engine, View   I
   E              Table,                E
   N          Compiler(C, C++, etc),       N
   T              ...                   T
```

```
DB 서버(AV5800)OS: DG-UX, DB: Informix 7.23UC1,
Memory:166MB HDD:22GB CPU: 256Mhz
```

Fig. 4. Environment of EEIS system

First of all, DB server contains Master DB. The internal client is used for internal data processing and external client is used for information request intermedium.

Also, VVS is used as interface between External client and DB server, and prevent DB server from overloading. As it has been stated before, there is no need for independent system for VVS. It can be implemented on any web server provided that it has enough disk space for DB engine.

The following configurations are used for VVS in order to provide the stability of the system. It describes user profile, DB access manager configuration file and Web Service and CGI associated files.

. **profile :**
INFORMIXDIR = /usr/informix
export INFORMIXDIR
PATH = $INFORMIXDIR/bin:/usr/webroot/usr-bin:$PATH
#INFORMIXSERVER = ncjav /* DB Server */
INFORMIXSERVER = cjst /* VVS Server name */

. **sqlhosts :**
cjst   onsoctcp cjst   sqlexec
ncjav   onsoctcp cjav   sqlexec

. **httpd.conf :**
ScriptAlias /cgi-bin/"/usr/webroot/http-docs/cgi-bin/"
<Directory "/usr/webroot/http-docs/cgi-bin">
    AlloOverrde None
    Options ExecCGI
    Order allow,deny
    Allow from all
</Directory>

3.2 Construction of a system

EEIS system can be described as in figure 5. It consists of Web server, DB server and VVS. They provides the information search capability, Information themselves, and control capability between Web server and DB Server respectively.
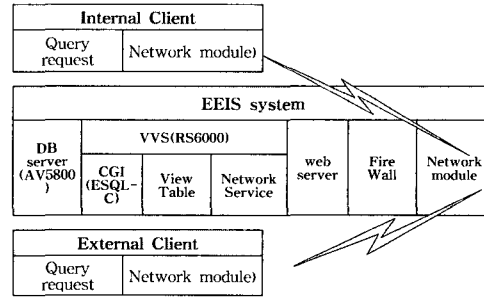

Fig. 5. Construction of system

3.3 Planning of a system

The current entrance exam results service System only provides keyword search service by utilizing the simple search key such as an application number, National ID number and/or Name in order to prevent it from network and DB server overloading as indicated by figure 6 and figure 7.

The main reason for considering the overload of DB server when it is used with Web server is due to the time spent to query and fetch results from DB server.
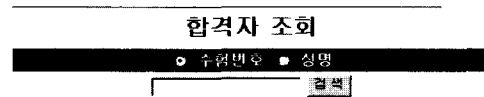

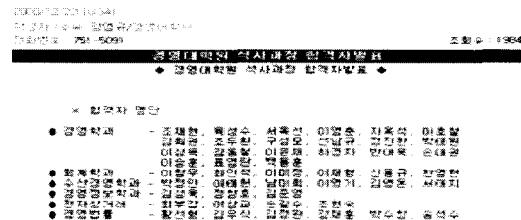Fig. 6. Search screen by key enter method

Fig. 7. Search screen of HTML

From the clients' perspective, there are two methods of fetching the information from DB Server. One is using static HTML document and the other is using DB server. When comparing two methods, the method of fetching data through DB server is more reliable for successful transaction and also provides the realtime data. However it takes more time related and other costs associated to fetch data through DB server. This is due to the fact that there is a need for an intermediary by using API functions provided by DB vendo.

In case of using DB API functions, it is required for CGI program to interrogate which DB it is from the DB server. And Based on this result, connect to DB, pass queries from DB clients to DB server, fetch query results from DB server to local memory and store the results. As each client repeats the above process, it accumulates and causes the overloading of DB server as well as degrading the service quality and access speed for the internal user.

This paper aims to resolve the problems described previously with EEIS system by employing the following strategy and VVS. When there are access requests from

multiple clients, the initial client for each individual data view would be permitted to access DB server directly. The following requests from the sub sequest clients for the data view would be directed to access VVS to get the requested data. This method would ensure the increased access speed for the internal clients as well as satisfying external clients via realtime data service such as search capability of number of prospective students for faculty and department, the merit lists for the prospective students.

3.3.1 Considering the Load of DB Server frame a policy

Figure 8 depicts the control algorithm for reducing DB server overloading. The user case would be depicted as follows. The multiple clients would request the realtime information through Web Server to DB server. The first client's request would be fetched from DB server and displayed his/her web browser. At the same time, the above result would be created as an HTML document at VVS. The same requests from the subsequent clients would be provided from the previously created HTML document from VVS instead of from DB server.

This user case would be applicable for cases such as requests from multiple clients for same prospective student, the resquest of confirmation of successful entry to University from a highschool faculty on their students. These multiple requests would always results in inherent overhead such as access to DB API functions and other essential process. It would also cause critical connection error to other part of Web and DB system as a whole.

In other word, the only request from the initial client for each data field would generate HTML codes on VVS. Meanwhile, the consequtive requests for same data field would be held off. Once the generation of HTML codes on VVS is finished, the other requests would be processed using the previously generated codes from VVS. EEIS control whole process automatically and prevent DB server from overloading.
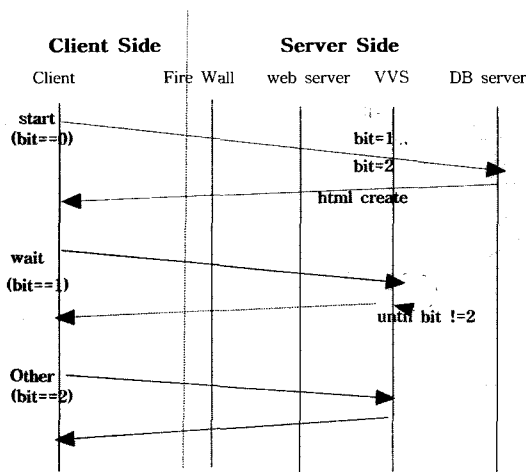

Fig. 8. Interaction operate protocol of client and EEIS

4. Implementation and evaluation

4.1 Implementation method of EEIS system

The proposed implementation method of EEIS system, which is described in figure 10, is different from the current implementation as described in figure 9 by placing VVS between DB server and Web server.

Also the security of DB server has been improved thanks to the access control ability implemented in CGI module of VVS system by isolating DB server from Web server.
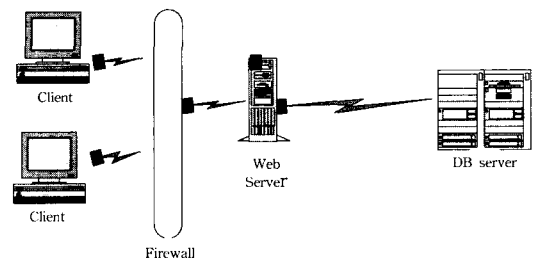

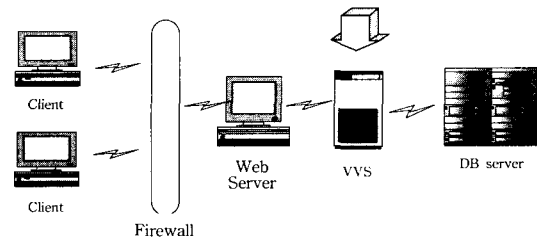Fig. 9. DB access method on internet

Fig. 10. DB access method of EEIS system

The main user interface of the new EEIS system has been depicted in figure 11. It shows the capability for the multiple fields of Data sorted and displayed by faculty and by school of the prospective students instead of the simple keyword such as a student name.
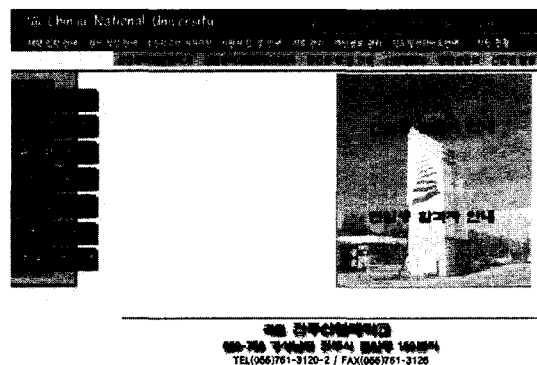

Fig. 11. Main menu of EEIS system

Fig. 12. Search result of EEIS System

4.2. Control of DB Server on CGI

As it has been indicated before, DB server access control algorithm is implemented within VVS system.

From the point of view of our approach for University Entrance Exam Notification System, the administrator set "io bit"of control file as initial state(0) while processing the successful candidates, additional accepted candidates and failed As the multiple clients would request a large amount of data for exam results for prospective students from DB server when university posts the results, the client would check the io bit when it is set as "zero". If the io bit has been initialised, then Client would set io bit as "one" and would try to connect to DB server. It is for the serialization of DB server access. That is to prevent multiple clients from attempting to connect to DB server simultaneously.

Once the data manipulation process has been completed in DB server, the "io bit"would be set as 2 indicating successful transaction of DB process. And it will be followed by generation of HTML codes in VVS for the subsequent access for the other clients.

If the "io bit"has been set as 1 when the other clients requests data from DB server, then it

4.3. Evaluation of system

The proposed method of implementing EEIS system has shown the significantly reduced time to fetch data for client from DB server compared to the current system. In some instances, the time has been reduced to 10 % of the time taken for current system.
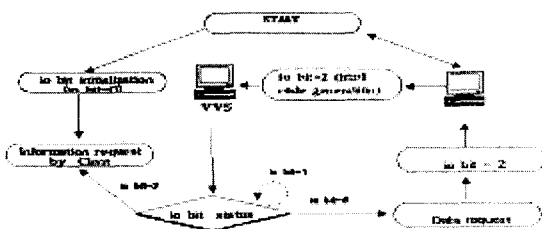

Fig.13. Control algorithm of DB server on CGI

Analysis of the new EEIS system has been conducted outside of normal office hours without indicates that the access attempt to DB server by the initial client has not been successful and/or the HTML codes generation process has not been completed. In this case, the client's requests would be blocked for a short while until the initial attempt has been

completed. Also, it has been shown that time to wait for this case is relatively short.

By using so, the subsequent queries by the other clients would be limited only by network load not by time to access DB server. It would also ensure the access to DB server for the internal users.
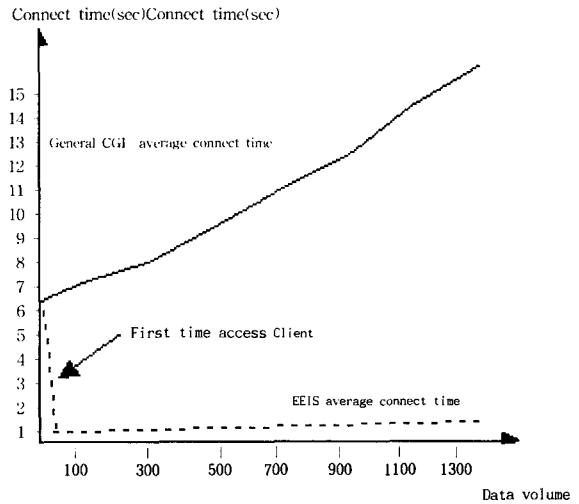

Fig. 14. Comparative analysis testing result of CGI and EEIS

The external DB server clients. The variables for the analysis include the number of clients and the number of requests.

The results include Client IP address, time of clients requests, time to complete client's requests.

According to the results, the time to access DB server through VVS in EEIS system has been reduced to ten fold compared to the current approach. It is similar to the time to access offline HTML documents.

As it has been shown in figure 14, the time to access DB server in EEIS system was less than 1 second for 100 300 requests. The current system spent 5 6 seconds for the same number of requests.

However, it is apperent that this result would not hold for the initial client requests or small number of requests.

5. Conclusion

The current system for University Entrance Exam result service over the internet is dependent on the simple keyword (name, the candidate number, National Identification number) search or static HTML document which is created by human intervention. Hence the results can be incorrect due to human error, and it also takes more time for University Administrator to provide services.

In order to correct these problems, the new EEIS system has been implemented with the consideration for DB server overloading as well as providing the realtime data more efficiently. It would also greatly improve the user satisfaction level by providing faster and various multiple query service.

Also it would remove the one of main reason for DB server overloading, and improve efficiency and the access speed for the internal clients. For the external clients, it would improve the user satisfaction and status of the University as a whole by providing the realtime and various information through EEIS system.

However, the proposed system implementation does not improve efficiency in case of small number of requests and/or the initial client query compared to the current system. Therefore, the further research in this area is required.

The current system for University Entrance Exam result service over the internet is dependent on the simple keyword (name, the candidate number, National Identification number) search or static HTML document which is created by human intervention. Hence the results can be incorrect due to human error, and it also takes more time for University Administrator to provide services.

In order to correct these problems, the new EEIS system has been implemented with the consideration for DB server overloading as well as providing the realtime data more efficiently. It would also greatly improve the user satisfactioin level by providing faster and various multiple query service.

Also it would remove the one of main reason for DB server overloading, and improve efficiency and the access speed for the internal clients. For the external clients, it would improve the user satisfaction and status of the University as a whole by providing the realtime and various information through EEIS system.

However, the proposed system implementation does not improve efficiency in case of small number of requests and/or the initial client query compared to the current system. Therefore, the further research in this area is required.

References

[1] 임종선, 주경수, "인터넷을 이용한 원격 학사정보 관리 시스템 개발", 한국멀티미디어학회 춘계학술발표논문집, 1998.

[2] Fabio Arciniegas 저, 류성열 역, "XML 개발자 가이드", 인포.북, 2001.

[3] 김승훈, 김성환, 김성동, 최기호, "XML(SMIL) 기반의 멀티미디어 메일 저작도구의 설계 및 구현", 한국멀티미디어학회 추계학술발표논문집, 1999.

[4] 김홍남, "CGI 파워 프로그래밍", 도서출판 대림, 1997.

[5] 최준형, 김상권, 정유현, "인트라넷 구축 및 활용방안, 1998.

[6] 이준석, "웹 브라우저와 CGI 프로그램 사이의 보안 통신을 지원하는 시스템 설계 및 구현", 한국정보처리학회 제6권, 1998.

[7] 김은경, 황병연, "WWW 데이터베이스 인터페이스를 위한 UCM(United CGI Management) 시스템의 설계", 한국정보처리학회논문지 제6권, 1999.

[8] KMK 정보산업연구원 편저, "SQL Server 7, 21일 완성", 삼각형프레스, 1999.

[9] Ed tittel, Mark Gaitjer, Sebastian Hassinger, Mike Erwin, 저, 권영진, 황인호 편저, "CGI 바이블", 영진출판사, 1997.

[10] Disan Petkovic 저, 문일보 역, "알기쉬운 마이크로소프트 SQL 서버 7", 정보문화사, 1999.

[11] Dan Connolly and Jon Bosak, "Extension Markup Language(XML))",1997,http://www.w3.org/XML/

[12] Jon Bosak, "XML, Java, and the Future of the Web, 1997, http://sunsite.unc.edu/pub/suninfo/ standards/ XML/why/xmlapps.htm

[13] Dan Connolly and Jon Nosak, "Extensible Mark-up Language(XML)", 1997, http://www.w3.org/XML/

Chang-Geun Kim

Chang-Geun Kim was born November. 20. 1962. He received the B.S degree in Department of Computer Statistic from Gyeongsang National University, Jinju, Korea, in 1985. He received the M..S degree in Department of Computer Science Gyeongnam University, Masan, Korea in 1991. He received Ph. D. degree in Department of Computer Science from Gyengnam University, Masan, Korea, in 1999. Since 1995, he has been a faculty member of the Department of Computer Science at the Jinju National University, where he is currently a associate Professor. His research interests are Neural Network. Fuzzy System, Computer networks, and Multimedia Communication etc. He is a member of KIMISC, KMS, and KIEE.

Phone : +82-55-751-3324
Fax : +82-55-751-3329
E-mail : cgkim@jinju.ac.kr

Byeong-Jin Choi

Byeong-Jin Choi was born December. 15. 1960. He received the B.S degree in Department of Computer Science & Engineering from Jinju National University, Jinju, Korea, in 2000. He received the M..S degree in Department of Computer Science & Engineering Jinju National University, Jinju, Korea in 2002. he has been a faculty member of the Academic information computer center(Computer information team) at the Jinju National University. His research interests are Neural Network. Fuzzy System, Computer networks, and Multimedia Communication etc.

Phone : +82-55-751-3182
Fax : +82-55-751-3187
E-mail : bjchoi@jinju.ac.kr

Han-Ho Tack

Han-Ho Tack was born july 6. 1959. He received the B.S. degree in Department of Electronic Engineering from Pukyung National University, Busan, Korea, in 1987 He received the M..S. degree in Department of Electronic Engineering from Dong-A University, Busan, Korea, in 1992. He received Ph. D. degree in Department of Electronic & Communication Engineering from the Korea Maritime University, Busan, Korea, in 1998. From 1987 to 1989, he was a Researcher at the Laboratory of Hung Chang Co. Ltd. Since 1991, he has been a faculty member of the Electronic Engineering at the Jinju National University, where he is currently a associate Professor. His research interests are Neural Network, Fuzzy System, Robotics, Factory Automation, Mechanical Vibration, Transfortation, and Multimedia System etc. He is a member of IEEE, KIMISC, KMS, KIEE, and KFIS.

Phone : +82-55-751-3332
Fax   : +82-55-751-3339
E-mail : fmtack@jinju.ac.kr