Performance Verification Process for Introduction of Open Source Software
-centered on introduction of Linux into the NEIS-

Doo Yeon Kim*, Jong Bae Kim**, Sung Yul Rhew***

Abstract Recently, introduction of Open Source Software into informatization of the government and public sector has been actively examined, however, Open Source Software is being rarely adopted due to the lack of verified and reliable data with regard to the criteria, process, performance and stability for introduction of Open Source Software.

In this paper, the process, method and plan for performance verification for introduction of Open Source Software into mission critical systems of the government and public sector are suggested in order to solve the aforesaid problem. Specially, a test system to judge whether or not to adopt Open Source Software in school affairs system of the NEIS(National Education Information System) of the Korean government was set up, and the method and process of performance verification by stage in addition to feasibility study were applied to the test system for verification. Based on the result of performance evaluation in the test system, the application of Linux to school affairs system of the NEIS is being successfully practiced. It is expected that this study will be a guideline to technical review process and performance verification method as necessary to introduce Open Source Software into the mission critical systems of government and public agencies.

Key Words : Open Source Software, Linux, NEIS, adoption, process, method, government

1. Introduction

It is a recent trend that introduction of Open Source Software is being allowed for with top priority as informatization is progressed by the government and public sector. However, in the mission critical system implementation projects of many public agencies, Open Source Software is being rarely adopted due to the lack of verified and reliable data with regard to the criteria, process, performance and stability for introduction of Open Source Software [1][2]. Stability and performance problem, if any, in the systems of public institutions including the government may fail services to the people and cause a huge waste of the nation's budget. It is the most serious worry of technical officials of government and public institutions to introduce Open Source Software. If a technical problem occurs after Open Source Software is introduced just through consideration of policy and then the system is implemented, they may be faced with several difficulties. Therefore the persons in charge cannot but be worried about the feasibility of what and how should be verified in advance to prevent all potential problems after the system is implemented.

This paper intends to suggest the process, method and plan for performance verification for introduction of Open Source Software into mission critical systems of the government and

* 교육인적자원부
** 숭실대학교
*** 숭실대학교 컴퓨터학과

public sector in order to solve the aforesaid problem. Specially, the method and process of performance verification by stage in addition to feasibility study are applied, and it is suggested as a case to adopt Open Source Software in school affairs system of the NEIS where the largest Open Source Software system is implemented in the worldwide public agencies.

2. Related Work

2.1 Open Source Software

Since starting with the Free Software campaign in 1984, Open Source Software has been up to the present time. Worldwide government agencies and private organizations have attempted the promotion of policy and the commercialization of Open Source Software since 2000 with a firm belief in the efficiency and development of Open Source Software, and the performance is being visualized[3][4][6][7][11]. Technical studies on the process and criteria for introduction of Open Source Software into the public sector are, however, not enough. The current technical studies mainly include policy data for introduction of Open Source Software or comparison of cost and performance between Open Source Software and commercial software.

2.2 Mode and Process for Introduction of Open Source Software

Introduction of Open Source Software is made in three ways: direct selection by the agency for introduction, indirect supply by referral from the outside such as private enterprise, and internal development and correction of Open Source Software by the agency[5].

Direct selection is advantageous in that it little incurs cost, while it is disadvantageous in that it has a huge risk burden such as responsibility for technical verification.

Indirect supply means that Open Source Software is supplied from an enterprise, which incurs technical support and service costs. Though it involves costs, it is so safe as to incur a less risk burden than direct selection.

In internal development, the most suitable software for the purpose of introduction is selected and downloaded among a lot of open software. It meansdevelopment and correction by the developer of the agency.

In public agencies, Open Source Software may be introduced by selection of one of the above-mentioned three means. When the Open Source Software industry and the current condition of the publicsector are considered, however, the most practicable and generally available way is indirect supply[5].

2.3 Problems and Considerations in Introduction of Open Source Software

Despite expectations of cost saving[8][9] and positive support of policy by the government, Open Source Software is not being actively introduced into mission critical systems. Most persons in charge of technology of the public sector hesitating the introduction of Open Source Software are of the same opinion that they cannot trust it because verified data about the criteria, process, performance and stability for introduction of Open Source Software are not enough. Also, they are not willing to introduce Open Source Software because they judge that systematic technical support, stable maintenance system, technical support on occasion of an operational problem and maintenance company, and tuning software are insufficient. However, it resulted from the misunderstanding and prejudice of Open Source Software, and it is objectively evaluated as below:

The first problem of Open Source Software is technical support. There is a users'complaint that they desire to receive technical support of Open Source Software but cannot easily find a support company. However, for Linux, typical Open Source Software, they may directly receive technical support from distributor, multi-national hardware vendor, and software developer. In addition, they may receive free support by referring to an Open Source Software community.

The second problem is reliability. One of the frequently asked questions about Open Source Software products is whether or not the quality of Open Source Software is satisfactory. The reason why such question is raised is the attribute of Open Source Software that the source is opened and purchasable without license fee. Like the exclusive software industry, diversified software systems are available for Open Source Software. It is known that the number of registered Open Source Software is 60,000~70,000, although it is not statistically exact. It is deemed that the fact that such Open Source Software is being adopted in actual services proves the reliability. According to a survey by Forrester Research for 140 US large enterprises in Oct. 2004, 53% of them used Linux as software operating system for their key services, and 52% of them had an intention to select Linux in introducing software into new services[10].

The third problem is performance and stability. It is generally recognized that Linux has less performance than Unix, however, seen from the results of benchmarks given from TPC-C[12] or SPECfp[13], Linux displays higher performance than Unix and Windows. In particular, it is being evaluated for its excellent stability in terms of operation at a level of operating system in domestic cases[5].

3. Plan for Performance Verification for Introduction of Open Source Software

In consideration of a general bias toward Open Source Software as pointed out in previous studies, suggestion of reliable data to change the recognition of the persons in charge of technology is necessary. The process, criteria, and method of performance verification for introduction of Open Source Software as well as all kinds of technical detail and supporting data shall be established before anything else. Thus this chapter is intended to suggest the process and method of verification and the plan for performance verification for introduction of Open Source Software, and describe a plan for performance verification to introduce Open Source Software by analyzing and arranging the results of applying it to implementation of school affairs system of the NEIS.

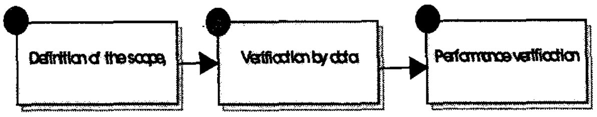
3.1 School Affairs System of the NEIS and the Performance Verification Environment

The school affairs system of the NEIS, an application case of this paper, is web-based system implemented with a view to promote the efficiency of educational administration and the provision of quality service to the nation. It deals with online the whole educational administrative affairs of 27 areas including school affairs, scholarship, human resources management, wages, accounting and civil applications in schools, offices of education, and the Ministry of Education & Human Resources Development by connecting 10,000 schools, 182 regional offices of education, 16 offices of educationin cities and provinces, and the Ministry of Education & Human Resources Development on Internet. Under the basic principle of implementing optimal system at lowest cost, Open Source Software was positively examined in this system.

The established school affairs system of the NEIS is operated by independent server or group server in 16 offices of education in cities and provinces. One independent server per school was constructed in special schools and high schools, while one group server per 15 schools in elementary and middle schools. Total 3,300 servers are composed in the target system; 2,338 independent DB servers, 606 group DB servers, and 393 WEB/WAS servers. Here, school affairs, admission and health care services of each school were operated, thenumber of users including teachers who were issued a certificate was 432,000, and the data of about 7 million students were processed into database. The maximum number of independent server users for test was 120, while that of group server users per 15 schools was 800.The test system for performance verification was composed same as that being actually operated in the schools. The DB servers consisted of one independent server, one group server, and WEB/WAS server. The independent servers had two CPUs and 4GB memory, the group servers had four CPUs and 8GB memory, and the system software was composed of Linux, Unix, DBMS and WAS as operating system. The requirements for adoption of application software are as mentioned below:

- The number of program modules for application capable of stably operating the service is 11,584, and the total number of database tables is 797(624 for school affairs, 64 for admission, and 89 for health care; one module means one file of JSP, JAVA, XML, mrd, jpg, html, properties, and jar).
- The Open Source Software shall secure performance and stability to deal with large-scale transaction centered on the time of concentration of services, and the arrangement on a specific time(end of the month, school term, and school year.).

3.2 Verification Process of Practical Plan for Open Source Software



<Figure 2> Verification Framework for Open Source Software

The verification framework for Open Source Software consisted of the definition of the scope and method of verification, data analysis, and performance verification as shown in Figure 2. According to this, the feasibility verification process for introduction of Open Source Software in school affairs system of the NEIS was divided into three stages. First, in the presence of an expert in Open Source Software, the scope and method of verification and the criteria definition statement were examined, applicability of the software was examined through technical verification such as close analysis on the function, performance and support system of an application server on the basis of data analysis and BMT(Bench Mark Test) and lastly, performance verification and optimization were conducted.

<Table 1> Verification Stages of Practical Plan for Open Source Software

Stage	Title of verification	Description
1	Definition of the scope, method and criteria for verification	Scope of verification - Linux operating system - Open Web/Was Server for Linux - Open DBMS for Linux Method of verification - Indirect investigation by data survey and analysis - Performance verification and optimization Verification criteria - Function and performance - Maintenance and support system

2	Verification by data analysis	Market trend Function and performance Support system Reference site Opinion of special institutions
3	Performance verification	Performance verification(response time, memory share etc.) in the actual test environment for Linux vs Unix operating system

For Open Source Software subject to priority verification, Linux was selected as operating system and Apache and MySQL as system software.

<Table 2> Open Source Software for Examination

Classification	Open Source Software for verification
Operating system .	Linux
System software	Web server(Apache) Web application(Tomcat) DBMS(MySQL)

Next, market trend, function and performance, support system, and reference site were established as the verification criteria for the examination of practicable plan for Open Source Software, and the details are as mentioned below.

3.3 Performance Verification for Server

As a result of primary review through data analysis on the basis of the aforesaid criteria and process, Linux operating system was included in the subject of performance verification as it was deemed to be available in terms of market trend, function and performance analysis and support system. However, the system software(Apache, Tomcat, MySQL) was not included in the subject of use of Open Source Software for the implementation of school affairs system owing to the inadequate connection between the required function and the system component.

<Table 3> Verification Criteria for Open Source Software

Classification	Description
Market trend	Whether or not to continuously build up the product functions according to the market trend of Open Source Software and the further trend of technologies Applicable to new system(school affairs system) through analysis on the main use of Open Source Software
Function and performance	Function and stability of Open Source Software Whether or not the major functions to support the performance and stability are fully equipped(connection pooling, clustering, security, DBMS transaction etc.) Whether or not to give support to standardization for design of open architecture Comparative analysis on the functions with commercial software Performance analysis by a reliable institution on the basis of the results of performance verification for Open Source Software
Support system	Current Open Source Software suppliers and whether or not they give technical support on a systematic basis Whether or not customer program is equipped against a legal dispute of Open Source Software
Reference site	Applications of Open Source Software by major agencies
Opinion of special institutions	Information policies and opinion of institutions specializing in information system Comment of institutions specializing in Open Source Software Opinion of institutions specializing in information protection

Next, the performance verification of server for mission critical services was conducted for Linux and Unix systems for design of the relevant architecture to implementation of school affairs system of the NEIS.

1) Implementation of performance verification environment

Performance verification intends not to evaluate the basic performance of hardware or software but to evaluate the performance of the mounted application software necessary for school affairs system. Accordingly, the scope of performance verification was limited to verification of whether or not transaction performance is met instead of verification in all types of Unix and Linux platforms. In addition, concerning the application and data for verification, the current NEIS application and sample data were used, and Mercury's Loadrunner[14] was used as measurement tool.

2) Criteria for performance verification and the scenario

Among school affairs of the existing NEIS, school record(inquiry, storing and closing) and monthly attendance management(inquiry, storing and closing) that are given most system load were selected. For performance verification, system response time and rate of CPU use were measured according to increase in the number of users, least items(resident registration number, name, ranking, class number) were encrypted centered on personal identification data, and performance difference was verified by such encryption. If the items of all data are encrypted beyond the need, it will cause a heavy burden on the system. The reliability, integrity and security of student data are secured by encryption of the identification information for individual student when they are stored in DB in order to prevent any other user from identifying them.

<Table 4> Performance Verification Scenario

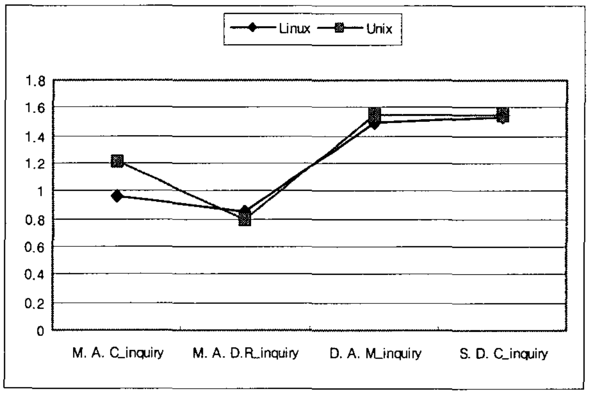
Independent server (Web/Was/DB)	Group serve		Encryption Applicatio
	Physical 2 Tier (Web/Was-DB)	Physical 3 Tier (Web-Was-DB)	
LX UX	UX-UX LX-LX LX-UX	UX-UX-UX LX-LX-LX LX-UX-UX	UX-UX LX-LX LX-UX

UX: UNIX, LX: LINUX

3) Results of performance verification

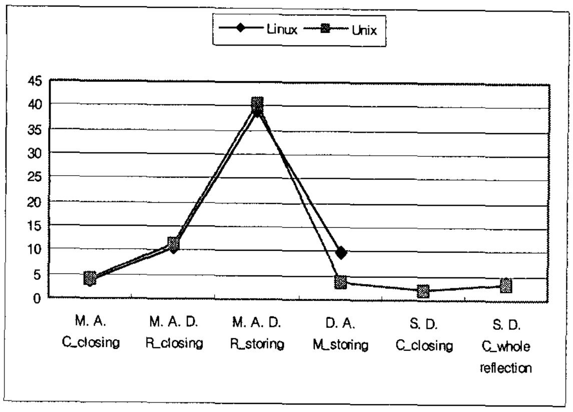
The results of performance verification are summarized as below:

- For independent server, response time was shown to be stable as less than 3 seconds with respect to monthly attendance closing, monthly attendance data registration, daily attendance management, and student department closing and inquiry in both Linux and Unix.(Figure 3.)



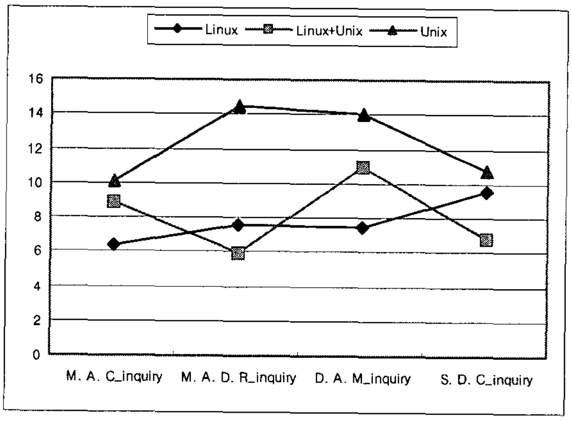
<Figure 3> Result of Performance Verification for Independent Server: inquiry

- For independent server, similar performance was shown in both Linux and Unix with respect to monthly attendance closing, monthly attendance data registration closing, monthly attendance data registration and storing, and daily attendance management and storing.(Figure 4.)



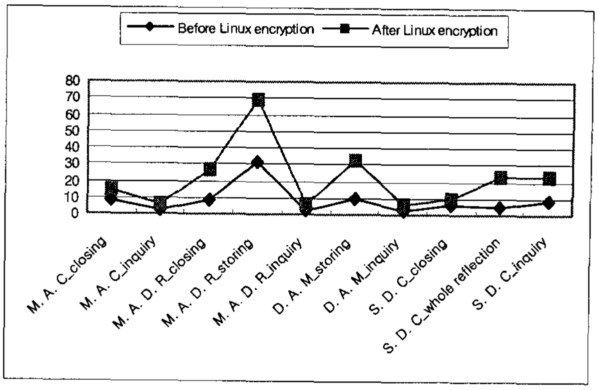
<Figure 4> Result of Performance Verification for Independent Server: closing

- For group server, the reference response time was satisfied with respect to monthly attendance closing, monthly attendance data registration, daily attendance management, and student department closing and inquiry in both Linux and Unix.(Figure 5.)



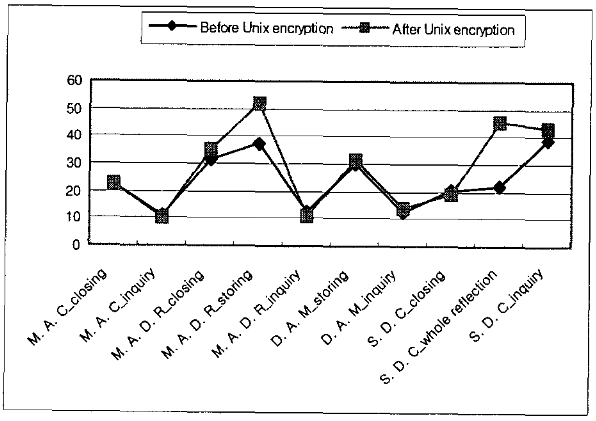
<Figure 5> Result of Performance Verification for Group Server: inquiry

- For group server, the response time to monthly attendance registration and storing was about 80 seconds in Linux and 50 seconds in Unix, which implies that it tended to be more stably changed in Unix as the number of users was increased.(Figure 6.)



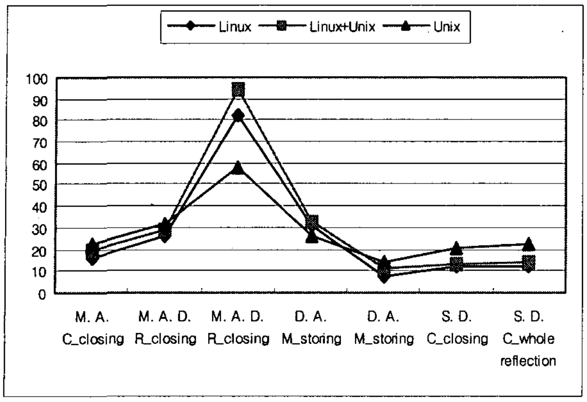
<Figure 6> Result of Performance Verification for Group Server : closing

- With respect to encryption of individual identification information, the response time was increased by about 70 seconds after encryption compared to about 30 seconds before encryption in Linux when monthly attendance storing and registration service was conducted. (Figure 7.)



<Figure 7> Changes in Response Time before/ after Encryption in Linux results of performance verification after update of the specification(2 cpu) -

- In case of encryption of individual identification information, little difference in the response time to monthly attendance storing and registration service was found between before and after encryption in Unix.(Figure 8.)



<Figure 8> Changes in Response Time before/after Encryption in Unix - results of performance verification after update of the specification(2cpu) -

The results of the aforesaid performance verification for independent server and group server are summarized as follows:

- For independent server, both Linux and Unix are available, however, when individual identification information was encrypted, Unix displayed more stable performance. The performance was reduced up to 2.5 times in Linux after encryption. There are two solutions to solve this problem; update of the specification or use of the 64bit server. Therefore enough test runs and operations are necessary to keep an optimal status in application of Linux server.
- For group server, as services were concentrated on a specific time in terms of the characteristic of school affairs, stable response time was displayed according to the number of users. Theuse of Unix server maintaining constant performance is also proper in encryption.
- System scalability shall be secured in preparation for addition of encryption items and increase of services(online service for university entrance examination, expansion of civil application services etc.) both in independent server and group server. As described earlier in the Section 3.1 performance verification environment, this performance verification utilized conventional school affairs application software of the NEIS and the sample data. Thus if it is applied with the application software to be newly developed for school affairs system and the practical data, there is a possibility that the estimated performance to deal with transaction may be increased, therefore it must be considered in actual introduction.

4. Conclusion

As the first stage, for implementation of new school affairs system in three areas including school affairs of the NEIS, the process and method of verification, the characteristics of service and the requirements for application software, and the selection and verification criteria for Open Source Software were established. Also, the applicability to actual services, the technical appropriateness and the stability were verified to look into the potential of introduction of available Open Source Software.

As the second stage, for design of the relevant architecture to implementation of the school affairs system of the NEIS, the performance verification of server (securing of basic data to estimate proper capacity of the server and feasibility verification of Linux) centered on mission critical services for operating system(Linux and Unix) was conducted. The performance verification was conducted by scenario of measurement after drawing out typical cases on the basis of the content of analysis on the current condition of services. The result was reflected upon design of the architecture of the school affairs system. The performance verification process was conducted in order of the implementation of performance verification environment, the preparation of the criteria and scenario for performance verification, the performance test, and the measurement of the result.

In this way, Linux, Open Source Software, was introduced into the pilot project for physical basis implementation as progressed together with the application software development project through a series of review after previous feasibility study and performance verification process prior to introduction of Open Source Software. The pilot project for school affairs system was conducted for 28 model schools and 108 test schools under the control of two offices of education in cities and provinces. Through the pilot project, whether or not to introduce Linux into independent DB server for this project under the physical basis was verified as the final stage.

The test operation environment, the performance criteria, and the content and result of the test after implementation of the test system for the pilot project of school affairs system are summarized as below:

As security system was mounted on the Linux-based school affairs system of the NEIS in model schools and the test run after encryption of individual student identification information met the criteria for both service handling speed and rate of resource use, Linux was finally adopted into about 2,300 independent DB servers in the nationwide physical basis implementation project for school affairs system. After the introduction of Linux into the independent DB server of school affairs system, optimization was conducted to display the best performance prior to opening of the system through performance verification and optimization (2 times).

As examined in the above, the introduction and operation of Linux, Open Source Software, into school affairs system of the NEIS involved a series of steps of data analysis, performance verification, test run, performance improvement and optimization. It is deemed that it is the most proper application model to introduce Open Source Software into the implementation of mission critical system in public agencies.It is expected that in the future, the method

<Table 5> Summary of Test Results for Open Source Software

Classification	Criteria
Test environment	- Performance test composed same as the actual operating environment - Subject : Independent/group server, WEB/WAS server - H/W : Independent server(2 CPUs, 4GB memory), group server(4 CPUs, 8GB memory) - Major system S/W : O/S (independent DB server :Linux, group DB server : Unix), DBMS (UniSql), WAS(JEUS) etc. - Application S/W : The part that school record inducing the largest load among the newly developed software applications is inquired and reflected is selected as item subject to test.
Performance criteria	- Maximum number of users in one high school : 120(14: simultaneous users) - Maximum number of users in 15 elementary and middle schools : 800(96 simultaneous users) - Response time : In 3 seconds on an average(except arrangement services) - Use of resources(CPU, memory) : 75% or less
Content of test	- Application test for the software developed and applied with respect to school record - Test of inquiry and reflection of school record for the student as well as list of students while adjusting the number of simultaneous users to 50%, 100% and 200% (independent server : 14, group server : 96)
Result of test	- After test, both response time and rate of resource use met the performance criteria. - Service handling speed : The response time satisfied the criteria; in 3 seconds on an average - Rate of resource use : The rate of CPU and memory use satisfied the criteria; both in 70% - System S/W : DBMS, Web/Was etc. satisfied both performance and stability.

suggested in this paper will be applied to the further introduction of Open Source Software into the mission critical services of public agencies, and there is a need to study detail modeling according to the subsequent matters to be replenished, the services of application, and the scale of system and project.

References

[1] Song Wee Jin, "The Characteristics of Technical Innovation for Open Source Software: Review"h, The Korea Technology Innovation Society Journal, Vol. 5, Issue no.2, pp.212-227, 2002.

[2] Korea IT Industry Promotion Agency, "eA Study on the Guideline to Introduction of Open Source Software"f, Dec. 2003

[3] Korea IT Industry Promotion Agency, A Study Report on Open Source Software, 2002.

[4] Korea IT Industry Promotion Agency, A Case Study on the Successful Introduction of Open Source Software, 2005.

[5] Ministry of Information and Communication, A Guide to Open Source Software, Jan. 2006

[6] Research, FLOSS(Free/Libre and Open Source Software: Survey and Study), FINAL REPORT, 2002.

[7] MICHAEL J. KARELS, "Commercializing Open Source Software", ACM Queue, vol.1, no.5, 2003.

[8] Al Gillen, Dan Kusnetzky, and Scott McLarnon, The Role of Linux in Reducing the Cost of Enterprise Computing, IDC White Paper, 2002.

[9] Robert Frances Group, Total Cost of Ownership for Linux in the Enterprise, 2002.

[10] Brad Day, Laura Koretzle, Linux Crosses Into Mission-Critical Apps, Gartner, 2004.

[11] Open Souce Initiative, OSI, http://www.opensource.org, 2006. 7.

[12] Transaction Processing Performance Council, TPC, http://www.tpc.org/tpcc/ 2006. 2.

[13] SPEC, Standard Performance Evaluation Corporation http://www.spec.org/results.html. 2006. 4.

[14] Mercury, http://www.mercury.com/us/products/performance-center/ . 2006. 4

[15] Linux, http://linux.org/ . 2006. 4

Doo Yeon Kim

- Regular member
- 1985 Seoul National Univ. fo Technology(B.E)
- 1998 Graduate School of the Soongsil University (M.E)
- 1997 Working as a Director of Educational Administration IT Division, Ministry of Education & Human Resource Development
- Interest : Software Reuse, Open-Source, Ubiquitous, IT Policy etc.

Jong Bae Kim

- Regular member
- 1996. 2 University Of Seoul (Management)
- 2002. 8 Graduate School of the Soongsil University (Master of Engineering)
- 2004. 8 Graduate School of the Soongsil University (Doctor of Engineering)
- Interest : Methodology, Open-Source, Mobile-Agent etc.

Sung Yul Rhew

- Regular member
- 1977 Soongsil Univ.(B.E)
- 1980 Yonsei Univ.(M.E)
- 1996 Aju Univ.(phD). After
- 1980 working as an professor in the Dept. of Computer Science, Soongsil Univ.
- Interest : Software Reuse, Process, Software Assurance, Maintance