Bringing Open Systems and Open Source Software to Large-Scale Enterprise Systems – for Better Business Computing

Chu J. Jong and Kyoungwon Suh
School of Information Technology, Illinois State University
Normal, IL 61790, U.S.A.
cjong@ilstu.edu, kwsuh@ilstu.edu

Abstract *- Open systems and open sources have made significant contributions to the information technology field. The importance of their implementation along with the number of IT shops using open systems and open software continue to increase. The growing number of open software challenges the traditional mainframe based enterprise systems which mainly host propriety applications. Mainframe engineers realized the inevitable trend of openness and started to accept open applications and implement open systems to host those applications. Moving toward the open systems and open sources not only reduces the cost of software development life cycle but also improves the quality of the service. However, there exists a gap between the mainframe system programmers and the open system programmers. To bridge the gap, we need to bring the openness to the mainframe computing field by educating both programmers the fundamental differences between the two systems. In this paper we present our efforts of bringing them together by integrating the open systems/sources courses into the mainframe computing education environment. Our initial attempts to bridge the gap include two open systems open sources course proposals and an internship plan for our Open-ECS development.*

Keywords: mainframe, enterprise computing, Open-ECS, ECS

1 Introduction

During the past two decades mainframes were announced to be dead many times and many people simply believed that the good old days of mainframe computing were gone. The history of mainframes [1] dates back to the 1940s. First-generation computers were made of vacuum tubes. In the 1950s and 1960s, transistor and multiprocessing technologies made evolution changes in the computer industry, which became dominated by mainframes. In the late 1970s, mini-computers such as the PDP-11 started to take a small portion of the computing market but that did not sustain long and quickly were replaced by microprocessor based workstations in the mid-1980s. In the 1990s, the Internet revolution further weakened mainframe systems. Most mainframe manufacturers ran out of resources and eventually abandoned their business. In the meanwhile, the mainstream computing industry moved to client-server based systems, followed by workstation based systems, and distributed cluster systems. Today, a few mainframe manufacturers exist and IBM is the one still actively producing improved mainframe systems [2].

Mainframes maintained their long popularity until 1980s because they not only offered better resource sharing but also provided maintenance free services to their clients. In users' mind convenience, security, and robustness are always high on their priority list. Mainframes are mainly designed for centralized computing environments in which users connect to their computing facility via a remote device. Information (applications and data) are stored in a central area to serve who logon the system. Users who have the access right to the mainframe computing facility will be able to perform tasks to accomplish their business objectives regardless when they need and where they are located. Mainframe users normally leave the supporting and maintenance jobs such as backup, security, recovery, upgrades, virus detections, and even printer cartage replacement to their centralized system administrator.

Mainframe systems cannot be easily replaced by other kinds of computing systems mainly because they are backed by many legacy business applications. According to recent studies, there are 200 billion lines of code written in COBOL that are currently used in today's business applications, with several billion lines of new code added annually [3, 4]. These applications, which are mainly used by major corporations for their daily business operations, are called legacy software and their hosting systems are called legacy computers (mainframes). The emerging commodity computing systems and the glories of Internet revolution forced mainframe systems to move out the mainstream computing. Mainframe programmers and technicians were quieted by the overwhelming fun stuffs such as parallel computing, internet applications, game development, web services, and so on. In late 1990's, blowing of the inflated internet bubbles brought people out of the hypes. People started to realize that a solid machine which runs forever is the one showing the real beauty, and mainframes continue surviving.

Though the demand for mainframe systems does not diminish significantly, we expect to fall short on the personnel who can develop applications for mainframes and maintain mainframes in near future. The studies of local and national Information Technology (IT) business indicate that current mainframe system programmers and system administrators are reaching retirement age; yet there does not exist younger programmer equipped with sufficient knowledge to fill the gap for operating mainframe systems [5, 6]. We expect that the demand to replace these employees will be high. In addition, a study of global computing markets shows that mainframe usage continues to rise. From both economic (e.g. power, space, installation, and maintenance costs) and business integration (e.g. centralized computing and storage system with distributed recovery strategy) aspects, the number of integrated mainframe systems will continue to grow in both major corporations and small to medium size businesses.

Although these are good indications that the old mainframe programming is coming back, we strongly believe that the new business computing systems will have a shape of mainframes integrated with open systems. The evolutional growth of IT already changed the way that people communicate and the manner in which they purchase goods and services. In particular, it has required businesses to adapt their operational strategies to fast changing IT trends. These phenomena have made a greater impact on transaction-based businesses, which use mainframes to conduct their daily operations. Based on our study of various mainframe computing paradigms, we believe that the integrated computing systems composed of mainframes, multi-platform servers and workstations, high-speed network, and heterogeneous storage devices will be widely used by enterprises to conduct their daily businesses. Based on this belief, we have developed a series of courses for the Enterprise Computing Systems (ECS) program at Illinois State University (ISU), which include two undergraduate and a graduate sequences, plus a graduate certificate program. The aim is to fulfill the demand of mainframe IT personnel, educate engineers on large-scale enterprise computing systems, and prepare students take the challenges of the growth of future integrated large-scale enterprise computing systems.

Starting at the end of 2004 the School of Information Technology (ITK) at ISU met with local and regional companies regarding how to address the demand, produce qualified engineers, and construct an integration strategy for the future. After more than three years intensive work, two undergraduate sequences, Enterprise Computing Engineering (ECE) for Computer Science majors and Integration Enterprise Systems (IES) for Information System majors, were approved and both of them started in fall 2008. Both graduate sequence and certificate program proposals also were approved by the graduate school. The recently approved graduate courses will be offered in fall 2009. Starting from fall semester 2006, ITK has offered three ECS courses: Introduction to Enterprise Computing Systems (three times); Operating, Data Communications, Networking, and Security of Enterprise Systems (three times); and System Programming and System Administration in Enterprise Computing Systems (twice). Online course offerings are also under consideration.

2 Objectives

The major goals of the ECS program are to fulfill the demand for mainframe IT personnel, to educate engineers on the Integrated Large-scale Enterprise Computing Systems (ILECS), and to develop the next-generation ILECS. An Integrated Large-scale Enterprise Computing System is made of a group of computing entities (including mainframes, servers, storages, and peripheral devices) which are interconnected by a network forming a virtual centralized computing facility. It is a computing system comprised of a set of computer technologies (hardware, software, and practices) used in integrated large scale systems. These integrated computing systems are mainly for transaction-based businesses and are widely used by service oriented enterprises for their business operations.

The goals of ECS program have been carefully examined to ensure their feasibility and deliverability. They should be implemented in a sequence of manageable steps. One step of the goal, educating engineers on the ILECS, is to emphasis the college education on the multi-platform computing systems integrated into mainframe based systems, in particular the IBM zSeries mainframe, for enterprises.

In order to integrate multi-platform systems into mainframe computing we need to bring the open systems and open sources into the mainframe enterprise environment, which is the main focus of this paper. We started with an education plan for the Linux operating system (the most popular open source operating systems) when we started offer our first ECS course in fall 2006. At the end of 2007 a Linux course development strategic team was formed, and later it became the open systems and open sources integration steering committee. The goal of this committee is to develop a curriculum for open systems and open sources in the area of mainframe-based enterprise systems. We named the course and curriculum development *Open-ECS*. The goal of Open-ECS is to oversee the future demand of integrated large-scale enterprise computer for better business computing. Main objectives of Open-ECS are:

i) to integrate the open system and open source courses into enterprise computing systems education,
ii) to ensure feasibility and deliverability of fulfilling the demand, and
iii) to study and develop an infrastructure of integrated large-scale enterprise computing systems.

3 Methodology and Approaches

From the past experience of developing the ECS program at ISU, we concluded that a successful course implementation should rely on strong collaboration among the three major domains, i.e., academic, business, and industry sectors. Establishing a joint effort among partners from these domains is the key to success. Identifying partners, establishing communications, and taking actions are the three components of the Open-ECS development.

3.1 Identifying Partners

The success of information technology is the result from improved products that lead to new applications. The new applications then drive new markets and new companies, ultimately increase competition and thus demand for better technologies. Economists call this a "virtuous circle" [6]. Major domains of this virtuous circle in our context are: academic institutions, business and software companies, and hardware manufacturers. Figure 1 depicts the roles and relationship between stakeholders of each domain. Our partners include Illinois State University and other colleges (the domain of academic institutions), State Farm, John Deere, Caterpillar, Discovery, and others (the domain of business and software companies), and IBM (the domain of hardware manufacture).

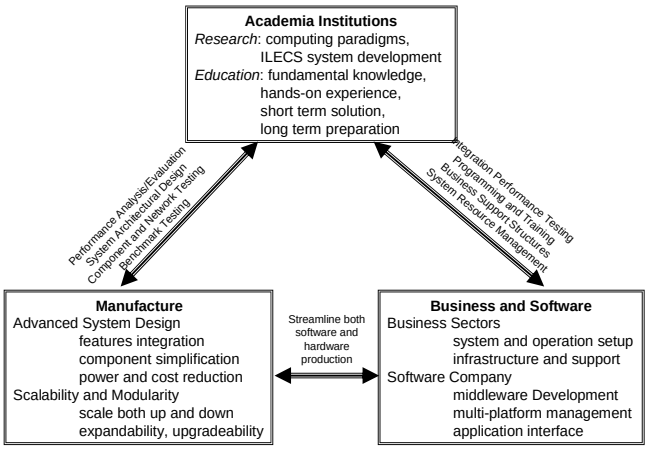

Figure 1. The relations between three stakeholders, manufacture, business, and academia (reprint from [1])

3.2 Establishing Communications

Open channels, contact points, and follow-up are the main factors to form a tightly coupled working group. By proper manipulation of these factors, our previous ECS curriculum development team worked together effectively. Thus we adopted similar strategy to build a stronger collaborative team among partners to facilitate the Open-ECS courses and curriculum development. We started to form a small core group to draft both course topics and implementation strategies and later on extended this group to a larger group to design detailed materials. The feedbacks on these drafts were used improve the revised course contents. Major revisions of the open systems/sources courses and implementation plan include integration, timelines, and supporting structures with a focus on the following items:

a. Sequence of open systems/sources courses.
b. Material to be included in each course.
c. Integration strategies and objectives.
d. Timeline and milestones to accomplish the objectives.
e. Fundamental supporting structures.

3.3 Taking Actions

We held regular progress meetings to exchange ideas and methods among partners. Follow-ups on the assigned action items keep partners to assume execution ownership and initiate new (or revise) actions. The actions are put into the following categories: open systems/sources integration strategies, cross platform education plan, material for education and training, internship rotation programs, virtualization implementation methods, networking strategy, and impact analysis.

To accomplish the Open-ECS objectives, we setup a four execution steps: Establishing an Open System Environment; Initiating Collaboration; Preparing Courses for Open System Integration; and Getting Student Involved.

Establishing an Open System Environment – We built a new open source integration environment based on our multi-platform parallel computing systems research lab, which is one of our research labs used by computer science upper level courses. The computing equipments available for both teaching and research of the Open-ECS include Sun SPARC Ultra and Blade workstations, Sun Enterprise servers, Pentium PCs, and Apple dual G5 Xserv. These systems will connect to the IBM z890 mainframe system at our university via a fiber optical switching network. By working with our manufacturer partner, IBM, we recently acquired a loaned Power system P520 to build course material for the AIX operating system.

Initial Collaboration – After a few initial contacts with our business partner, State Farm in 2006, we held a kick-off meeting in December 2007 to establish a joined effort for Unix/Linux education development. Since then, several meetings were held in May, July, and September 2008. Subsequently IBM joined the development meeting in October 2008. Prior to these meetings, we met with John Deere twice to develop system infrastructure and integration case studies. We present a green computing [8] in the 7th ECS program status update briefing on March 29th 2008, and we also conducted several ECS Initiative (ECSI) meetings among ITK faculty and staff. The feedbacks from the meetings indicate that integrating the open source with mainframe enterprise computing should be the next step for us to pursue.

Preparing Courses and System Integration – A few open system integration topics under consideration are: Unix/Linux System Administration; Advanced Unix/Linux System Administration for Enterprise Computing Systems; and Open Systems and Large-scale Enterprise Computing Architecture. Some open source integration topics such as Enterprise Web Server, Web Application Organization, Enterprise Database Systems, and Service Oriented Architecture are also under consideration for the future Open-ECS development and implementation. We aim to combine experimental education with traditional research activities in order to produce problem-solving oriented system engineers.

Student Involvement – An important part of the multi-lateral education approach [8] involves both traditional students and non-traditional students (working professionals) in the course setting. Our experience suggests that topic discussions among these students not only increase their analytical skills but also stimulate their logical construction ability. Combining experimental working knowledge with learned theoretical methods will enhance their problem solving capability which is a crucial hands-on skill toward future success. For example, we had one student worked on optical communication project in spring 2008, another student started Linux networking project in fall 2008, and a couple of students have been working on z/VM running SuSE Linux since spring 2009. From these projects students may discover that although a simple communication between two entities seems easy and a well known application implementation seems straightforward, the actually execution details are challenging. Also, these projects provided us with precious feedback on teaching and learning information to enhance future course delivery.

4 Draft Timeline and Deliverable

It usually takes six to twelve months to prepare new courses after outlined their objectives in ISU. The preparation processes include: identifying the audience, designing course proposals, setting up teaching and lab facilities, building course contents, and advertising the courses. The draft time-line and deliverable estimation of two pilot courses present in the following were from the summer 2008.

Summer 2008 – Establish the Open System Integration Lab and install open system software for both teaching and research. This lab is located next to the Enterprise Computing Systems lab where the z890 system can be accessed via a fiber optical switch.

Fall 2008 – Conduct collaborate work with State Farm, John Deere and others (more being expected) to define the skill set requirements. Collaborate with ITK/ISU faculty and staff to get ready to establish new courses. Prepare the first pilot course, Linux/Unix System Administration.

Spring 2009 – Deliver the first pilot course and prepare new course proposal to be approved by the curriculum committee.

Summer 2009 – Prepare the second pilot course, Advanced Unix/Linux System Administration for Enterprise Computing Systems, and ready to deliver in fall 2009.

Fall 2009 – Deliver the second pilot course and ready to submit another new course proposal.

5 The Outcome

After meeting with State Farm in regard to the Linux/Unix education for two years, IBM Power System group joined us in October 2008. By then, we had already started the planning for the Open-ECS and the Linux/Unix education becomes part of the Open-ECS. After October 2008, we added AIX operating system to the Linux/Unix setting and produced two course outlines. To encourage students' participation and increase hands-on exercise, we developed an internship program. These are the result of collaborated efforts by the three domains as shown in Figure 1. Stakeholders are from Illinois State University, State Farm Corporation, and International Business Machine Corporation. The following are the descriptions of the work we have completed during the past eight months.

5.1 Open Operating Systems for ECS

This section presents the open system development plan and time-line schedule, a part of the Open-ECS. Following the development plan are outlines of the two courses, Introduction to Linux/Unix-AIX Operating Systems and Enterprise Linux/Linux-AIX System Programming and System Administration. Due to the delay of the arrival of Power System, the planned schedule needs to be pushed four months.

We have consulted many Linux/Unix text book, training references, and self study guides to construct the outlines of the two courses in section 6.1.2 and 6.1.3. Contents of some books [10, 11, 12] have been adopted in large and some of them with similar material were omitted. The material related to AIX operating system are mainly from IBM workshop training [13] and there will be more supplemental material from IBM redbooks (AIX) and system manuals.

5.1.1 Open O/S for Open-ECS

ISU/State Farm/IBM Linux/Unix Enterprise Computing Systems Course Development Proposals

***Objective*:**

Complete a series of two Linux/Unix-AIX courses in line with the Enterprise Computing Systems program at Illinois State University (ISU).

***Audience*:**

College students and working professionals who are seeking fundamental and/or advanced knowledge of open source Linux and Unix-AIX operating systems. In addition to learning the system programming and system

administration, Open Source and mainframe computing systems integration will be addressed.

Two Courses:

I: Introduction to Linux/Unix-AIX Operating Systems.

II: Enterprise Linux/Unix-AIX System Programming and System Administration.

Facility:

Text book, lab menu, and supplemental documents will be compiled by the faculty of School of Information Technology (ITK), ISU. Lab equipments, such as PC, small workstations will be provided by ITK for Linux installation. Other equipments, such as IBM Power system series will be supplied by IBM for Unix-AIX and/or Linux installation. In the following tentative time-line schedule, we assume the IBM Power system will be ready prior to start developing the lab assignments.

Tentative Time-Line Deliverable:

Course I

Compiling Material:	10/2008 to 03/2009
Propose Pilot Course:	01/2009 to 03/2009
Developing Lab:	01/2009 to 04/2009
Implement/Testing:	03/2009 to 05/2009
Delivery Course:	
Training:	June, 2009 (40 hours training)
Classroom:	Fall 2009

Course II

Compiling Material:	04/2009 to 10/2009
Propose Pilot Course:	09/2009 to 11/2009
Developing Lab:	07/2009 to 11/2009
Implement/Testing:	10/2009 to 01/2010
Delivery Course:	
Training:	January, 2010 (40 hours)
Classroom:	Spring 2010

5.1.2 Outlines of Course I

Introduction to Linux/Unix-AIX Operating Systems

1. Overview – What is Linux (Unix-AIX)
   - 1.1 Introduction to Operating System
   - 1.2 Linux History
   - 1.3 Getting Started
     - 1.3.1 Computer Hardware
     - 1.3.2 Linux Software Architecture
     - 1.3.3 Log On/Off
2. Shells and Text Editing
   - 2.1 What is shell
   - 2.2 Various Linux Shells
   - 2.3 Shell Basic
   - 2.4 Creating Text File
     - 2.4.1 Full screen editors – pico, vi, Emacs
     - 2.4.2 Graphical Editor -- XEmacs
3. Electronic Mail
   - 3.1 Basic e-mail concept
   - 3.2 Using mail command
   - 3.3 Full Screen e-mail systems – elm, pine
   - 3.4 Graphical e-mail system – Kmail
   - 3.5 Other e-mail systems
4. Linux File Systems
   - 4.1 Types of Files
   - 4.2 File System Structures – Attributes and Directories
   - 4.3 Storage – inode
   - 4.4 File Security
   - 4.5 File Archive and Backup
   - 4.6 Differences between Linux and Unix-AIX File System Structures
5. File Processing
   - 5.1 Fundamental – viewing, copying, moving, removing, appending, combing, comparing, and printing, … etc.
   - 5.2 Advanced – regular expression, sorting, searching, encoding, decoding, etc.
   - 5.3 File Sharing
6. Redirection and Piping
   - 6.1 I/O redirection
   - 6.2 stdin, stdout, stderr
   - 6.3 pipes
7. Processes
   - 7.1 Multi-processing and Scheduling
   - 7.2 Process States
   - 7.3 Process Attributes and Job Control
   - 7.4 Process Hierarchy
8. Networking and Inter-connection
   - 8.1 Computer Network and Internetworking
   - 8.2 Network Models and Protocols
   - 8.3 Network Applications
9. Shell Processing – bash and TC shells
   - 9.1 Shell variables and commands
   - 9.2 Passing Variables to Shell Scripts
   - 9.3 Program Control and Flow Control
   - 9.4 Numeric Data Processing
   - 9.5 Array Processing
   - 9.6 Signal Processing
   - 9.7 Debugging Shell Program
10. Tools for Software Development
    - 10.1 Programming Languages
    - 10.2 Compilation Process
    - 10.3 Make File
    - 10.4 Program Tools
    - 10.5 Analysis Tools
    - 10.6 Debugging and Performance Tools
11. GUI
    - 11.1 Xfree86
    - 11.2 GNOME
    - 11.3 KDE
    - 11.4 Others

5.1.3 Outlines of Course II

Enterprise Linux/Unix-AIX System Programming and System Administration

PART A) Install and Setup Linux Systems

1. Linux Distribution and Hardware Platform Hosting System
   - 1.1 Install and re-compile kernel
   - 1.2 Install and porting applications
   - 1.3 System Configurations and Loadable Modules
   - 1.4 Unix-AIX Booting, Installation, Setup, LPAR, and WPAR
2. Basic Administration
   - 2.1 Manage Users
   - 2.2 Manage Applications
   - 2.3 Manage System Resources
   - 2.4 Single System Backup/Recovery
3. Administration Tools
   - 3.1 Regular Expression
   - 3.2 Perl Programming
   - 3.3 Other Scripting Languages
4. Advanced System Administration
   - 4.1 Other Scripting Languages
   - 4.2 Kernel Basics and Tunable Parameters
   - 4.3 System Library Structures
   - 4.4 Service, API, and Run Scripts

PART B) Linux (Unix-AIX) Clustering and Networking

5. Networking Fundamentals
   - 1.1 Other Scripting Languages
   - 1.2 Protocol and Network Topology
   - 1.3 TCP/IP
   - 1.4 Intra and Inter Networking
   - 1.5 VPN
6. Network Administration
   - 5.1 Manage Users
   - 5.2 Manage Applications
   - 5.3 Manage System Resources
   - 5.4 Network File Systems
   - 5.5 Backup and Recovery
7. Network Administration – Tools and Performance Monitoring/Improvement

PART C) Enterprise Computing Linux and AIX

8. z/OS and zArchitecture Basic
9. Virtual Machine Fundamentals
10. z/VM and zLinux
11. Enterprise System Networking
    - 10.1 Intra Node (LPAR) Communications
    - 10.2 Inter Node (LPAR) Communications
    - 10.3 Coupling Mechanisms

11. Web Application and Middleware

5.2 Student Involvement – Internship at ITK

In Section 3 we stated that student involvement contributes in large to the teaching material and classroom settings. To engage students, we started to offer summer internship and semester research programs in ITK for John Deere Corporation since 2007. Although this is a company sponsored program, students work in our university site rather than company site. ITK faculty is responsible for the research assistant program and ITK faculty work with university's Administration of Information System (AIS) department to manage the summer interns. Using similar approach, we proposed an internship program to State Farm for the students who are interested in the Unix/Linux-AIX development. The following is the internship and research program proposal.

ISU State Farm Linux ECS Student Intern and Research Program Proposal

Objective:

Establishing Linux computing system administration infrastructure to provide intern and research opportunities to students at the School of Information Technology (ITK) at Illinois State University (ISU).

Description:

The infrastructure of the Linux systems to be established at ITK will provide an integration environment for both mainframe computing systems and Linux open source systems to form an Integrated Large-scale Enterprise Computing Systems (ILECS). This will enhance the Enterprise Computing Systems (ECS) program at ISU and fulfill the current and future demand of IT professionals. Students need to acquire the open system knowledge to fit in the ILECS environment and conduct business operations. The goal of the internship program is to generate a pipeline to recruit students to the ECS program, starting from their sophomore year to senior year. The focuses will be hands-on training, practical education, system administration, and topic exploration. The goal of the research assistantship program is to enhance both software and hardware infrastructure of ILECS and the focuses will be research the system administration, system integration methods and strategies, and advanced specialized areas. To become a research assistant of these areas, sufficient computer knowledge is essential and thus the target students are senior to graduate. The detailed plan will be completed in a separate document upon approval of this proposal.

Collaboration:

Illinois State University – ITK will provide environment for both internship and assistantship during the school semesters, that include duty assignment, topic direction, and working facilities.

State Farm Corporation – State Farm Corporation will be the program sponsor in terms stipend, travel, and administration cost. A budget plan will be completed upon approval of this proposal. During the summer time, State Farm may invite students to work with them as their summer co-op.

Two Year Projection (State Farm funding will be noted):

Fall, 2009 – one interns and one research assistant.
Spring, 2010 – two interns and one research assistant
Summer, 2010 – one research assistants, presentations
Fall, 2010 – three interns and one research assistant
Spring, 2011 – three interns and one research assistant

Summer, 2011 – one research assistant, presentations

6 Future Impact and Conclusion

The globalized business paradigm results new service delivery infrastructures. Traditional computing facilities face the challenge of interoperability and thus require another computation layer add on top of existing platforms. With this middleware layer, seamless communication between applications can be achieved and the interoperability can be solved. The open system computing provides combination of interoperability, portability, and the open software standards will bring independent platforms together. The commonly agreed development environment based on open software standards provides better facilities for developers to contribute their intelligence without hurdler through platform specific issues. Integrating open source into large-scale enterprise systems soon will become the mainstream of business computing, thus bring the competence level up for graduates is a critical factor to the success of future transaction-based computing.

It takes years of collaborated efforts to start the ECS program and it has been years since we started the initiation of the open system course of the ECS program. With all the work we have contributed so far, we have not implemented the first Open-ECS course yet. A series courses and a complete curriculum require deliberate thinking and planning. Is the future enterprise computing systems already been defined or waiting for us to create? This is a profound question and no one can answer it without knowing what the future will be. Bring the openness to enterprise computing, the Open-ECS is just the beginning of the endeavor from the partners of the three aforementioned domains. Working together toward the future enterprise computing we might be able to delineate tomorrow's environment with better confidence.

7 References

[1] C. Jong, *A Study of Mainframe Computing System to a Degree Program*, in Proceedings of International Conference on FECS 2007, Las Vegas, Nevada, June 25 – 28, 2007

[2] http://en.wikipedia.org/wiki/Mainframe/

[3] E. Arranga and F. Coyle. COBOL: Perception and Reality, Computer, Vol. 3, No. 3, March 1997

[4] N. Stern and R. Stern. Structured COBOL Programming, 9th Edition, John Willy & Sons, New York, 20

[5] P. Thibodeau. Shortage of mainframe skills may give IT execs gray hairs, Computerworld, IDG Network, 2005.

[6] D. Carr and R. J. Kizior. The Case for Continued Cobol Education, IEEE Software, 2000

[7] A. Tanenbaun. Structured Computer Organization, fifth edition, Pearson Prentice Hall, 2006

[8] http://www.itl.ilstu.edu/ecs/GreenComputing.ppt

[9] C. Jong and C. Li. Multi-Lateral Constructive Learning Process with Hands-On Problem-Based Solving Technology, in proceedings of FECS2008, WORLDCOMP'08, Las Vegas, July 14 – 17, 2008

[10] Syed M. Sarwar, Robert Koretsky, and Syed A. Sarwar. Linux: The Texebook, Addison Wesley, 2002

[11] Evi Nemeth, Grath Synder, and Trent Hein. Linux Administration Handbook, 2nd Edition, Prentice Hall, 2007

[12] Harley Hahn. Guide to Unix and Linux, Mc Graw Hill, 2009

[13] IBM Training. AIX for UNIX Professionals Workshop, April 28 – 30, Chicago, IL

[14] http://www.redbooks.ibm.com/