INTEGRATED PEOPLE-ORIENTED DEVELOPMENT SCHEME
- To improve IT efficiency by centralizing the management of user authorization for tertiary education institution

BY
YIP CHI HONG

Presented to the Faculty of Business
City University of Hong Kong in Partial Fulfillment of the Requirement for the Degree of

Master of Business Administration

CITY UNIVERSITY OF HONG KONG
2009
Executive Summary

The Department of Computer Science (CS Dept) is one of most IT-intensive departments in City University of Hong Kong (CityU) which prompts the innovative technology for teaching, learning and researches. With its rapid growth over the past 20 years, CS owns many in-house well developed administrative systems that are all maintained by CS Laboratory (CS Lab). At present, these legacy systems have raised lots of problems relating to the user account management of CS Dept. In addition, it leads the impact on existing limited labor resources if this situation remains unchanged. Therefore, it raises the management concerns and wants to resolve the present unproductive workflow by simplifying the administrative role-based assignments across different application systems in CS Dept.

A system integration development proposal called “Integrated People-Oriented Development Scheme”, which code name is IPODS, is to centralize all user access control management and authorization under Role-Based Access Control (RABC) framework. In order to meet the high service level requirements in CS Dept, IPODS is designed to align with the modularized structure of Service-Oriented Architecture (SOA). In addition, IPODS applies the open standard protocol by eXtensible Access Control Markup Language (XACML) to deal with data access control management and encounter with data exchangeability at the same time. In comparison with alternative project proposals, IPODS is evaluated to achieve higher ratings on the five IT business
objectives of CS Dept, such as *Cost of Ownerships, Dependence, Demand Growth, User Impacts* and *Usage Frequency*, in overall.

Since the implementation phase of IPODS project doesn’t match the timeline of MBA project, it is limited to evaluate the project result by simulation. The simulation result indicates that it gains 5 times efficiency on the role-based assignment process. Besides, it is predicted that CS Dept will have a streamlined, robust and auditable access control management system to achieve better system integration flexibility and higher user service level standards. These intangible qualities can be measured by IT effectiveness metrics on the different dimensions, such as efficiency, performance, productivity, and user satisfaction.

In conclusion, IPODS brings out an implication on the business application design about the importance of platform openness and system modularity to achieve higher reusability and service pledge in business organization. Furthermore, it has high potentials to generate several opportunities, such as *Separation of Duties, Role Optimization, Automatic Account Replication* and *XML Account Portfolio*, for continuous improvement. Therefore, IPODS is worth for implementation so as to maximize the benefits of IT support management in CS Dept.
Acknowledgment

Although I am the only one who is responsible to take in charge of this project, it actually is the teamwork of many people around me. With their helps and encouragements, I successfully complete the whole project in final, so I would like to take this opportunity to thank them.

First of all, I have to express my great appreciation to my supervisor, Dr Yulin Fang, for his kind guidance to transform this project from the focus of technical view to be more business-oriented approach with managerial insights over the boundary of my working position. With his valuable recommendations, I understood more about the present situation of the case within my working organization and how to utilize some information system methodologies to justify a problem and resolve it finally.

I also like to thank all colleagues in CS Lab, especially Mr. C M Hui, who provided lot of supporting information about system infrastructure, which is useful for my project studies. More importantly, he always inspires me to elaborate my solution every time when I have difficulties.

Finally, I thank to my wife for taking care of my two beloved sons, so let me have sufficient time for my MBA studies.
# Table of Content

Part I: Case Study ......................................................................................................................... 9  
  i. Introduction of Department of Computer Science ................................................................. 9  
  ii. Organizational Structure ...................................................................................................... 13  
  iii. Overview of IT Support ........................................................................................................ 16  
  iv. Present Situation and Business Objectives ......................................................................... 19  

Part II: Case Analysis .................................................................................................................. 23  
  Chapter 1: Problem Identification ............................................................................................ 23  
    1.1. Problem Statement ........................................................................................................... 26  
    1.2. Project Mission ................................................................................................................ 27  
  Chapter 2: Literature Reviews .................................................................................................. 29  
    2.1. Service Oriented Architecture (SOA) .............................................................................. 29  
    2.2. Role-based Access Control (RBAC) ............................................................................... 31  
    2.3. eXtensible Access Control Markup Language (XACML) ............................................ 35  
  Chapter 3: Managerial Justifications ....................................................................................... 37  
    3.1. The Trend of Integration Approach .................................................................................. 37  
    3.2. Project Priority Assessment ............................................................................................ 41  
  Chapter 4: Analysis and Findings ............................................................................................. 43  
  Chapter 5: Design & Implementation ....................................................................................... 47  
    5.1. Conceptual Design ............................................................................................................ 48  
    5.2. Implementation of XACML Architecture ...................................................................... 50  
    5.3. Project Plan ...................................................................................................................... 52  
    5.4. Technical Specifications .................................................................................................. 54  
    5.5. Project Team .................................................................................................................... 55  
    5.6. Project Lifecycle .............................................................................................................. 56  
    5.7. Implementation Methodologies ........................................................................................ 57  
    5.8. Testing Approaches ......................................................................................................... 59
5.9. Deployment & Continuity Planning ................................................................. 60
5.10. Project Risk Assessments ........................................................................ 61
Chapter 6: Evaluation of Implementation .......................................................... 62
Chapter 7: Prospects for Further Improvement .................................................... 65
  7.1. Separation of Duties (SoD) ....................................................................... 65
  7.2. Role Optimization .................................................................................... 65
  7.3. Automatic Account Replication ............................................................... 66
  7.4. XML Account Portfolio ............................................................................ 67
Chapter 8: Summary and Conclusions ................................................................. 68
  8.1. References ............................................................................................... 70
  8.2. Appendix A: Simulation with 1% error rate ............................................. 73
  8.3. Appendix B: Simulation with 5% error rate ............................................. 74
Part III: Self Evaluation ....................................................................................... 75
  i. The Core Value of Case Study .................................................................... 75
  ii. More Understanding about Information System ....................................... 76
  iii. Difficulties on Project Evaluation ............................................................ 77
  iv. Implications on System Openness & Modularity ..................................... 78
List of Figures

Figure 1: Organization Chart of CS Dept ................................................................. 13
Figure 2: Overviews of IT Support Management in CS Dept .............................. 17
Figure 3: The business objectives of software development projects in CS Dept. ... 22
Figure 4: The Workflow of Web Application Login ............................................ 23
Figure 5: 3 RBAC Methods for CS Web Applications ........................................ 24
Figure 6: The Overview of Integrated People-Oriented Development Scheme ....... 28
Figure 7: Relationships between role and permissions on RBAC0 ...................... 32
Figure 8: Relationships between role and permissions on RBAC1 ...................... 33
Figure 9: Implementation Design of XACML ....................................................... 35
Figure 10: Application Pool in CS Dept before 2003 ......................................... 37
Figure 11: Application Pool of CS Dept in the period between 2003 and 2005 ...... 38
Figure 12: Application Pool of CS Dept planned after 2010 .............................. 39
Figure 13: Workflow comparison between the present and planned RBAC ........... 45
Figure 14: Paradigm Shift of Access Control Management Model .................... 47
Figure 15: Overview of RBAC Matrix ................................................................. 48
Figure 16: State Diagram of RBAC Operations .................................................. 49
Figure 17: Dataflow Summary of XACML ............................................................ 51
Figure 18: Development Project Lifecycle in CS Dept ........................................ 56
List of Tables

Table 1: Benefits of RBAC Implementation ............................................................... 33
Table 2: Project Priority Assessment Table ............................................................ 41
Table 3: Forecasted performance gain by simulation ........................................... 46
Table 4: Project Schedule of IPODS .................................................................. 52
Table 5: Infrastructure Requirements for IPODS ................................................. 54
Table 6: Switching Cost of Project Development ............................................... 61
Part I: Case Study

i. Introduction of Department of Computer Science

The Department of Computer Science (CS Dept) in City University of Hong Kong (CityU) was found in 1984 as the same year as City Polytechnic established. It launched its first honor degree course of Computer Science in 1987 and then its first PhD graduate in 1994. From traditional fundamental courses such as computer architecture and software engineering to advanced specialized courses like object-oriented distributed databases and webcasting, CS Dept provides a wide range of different course curriculum to cover the major 4 subject groups on Software Engineering, Distributed Networking, Mobile Communications and Computer Graphics. On the applied research areas, CS academic staffs have significant contribution on real-time image computing, computation algorithms & its theories, and artificial intelligence.

Although CS Dept hasn’t stated any explicit mission statement on any public communication channels, the head message published on the CS departmental website (Yao, 2006) provides some ideas and implications. From my interpretation of CS Head’s message, CS Dept aims to provide a comprehensive education for the graduates to equip themselves with technical skills, expert knowledge and professional ethics which are capable to tackle the challenges and even plays leadership roles in the information age. In align with CityU mission, CS Dept is to offer professional computer science oriented programmes for students to achieve higher quality tertiary education of
international standing in the Asia-Pacific Region. Moreover, its objective is to prepare undergraduate and graduate students for productive careers in industry, academia and government by providing environment for teaching, learning, and researching in the applications of computing. To enhance the quality of higher education, CS Dept also places high priority on the establishment of innovative application research environment and make it as important resource center for discovering, integrating and applying new knowledge and technologies.

In CS Dept, we have 6 teaching courses at undergraduate and postgraduate levels:

- **BSc (Hons) in Computer Science**
- **BSc (Hons) in Computer Studies**
- **BSc (Hons) in Creative Media**
- **MSc in Computer Science**
- **MSc in Electronic Commerce**
- **Postgraduate Certificate in Information Security**

On the coming year, it also prepared to launch 2 new courses:

- **BEng (Hons) in e-Logistics and Technology Management**
- **BA (Hons) in Digital Media Broadcasting**

According to the staff directory published on 1st November 2008, CS Dept has 47 academic staff, 78 research staff, 56 research students, 9 technical staff, 10
administrative staff, around 500 students and more than 100 new alumni every year. In addition, CS Dept has managed 5 major laboratories:

**Computer Science Laboratory:** it owns a large number of servers including SUN E220R and E4800 Enterprise Servers, over 300 workstations and PCs which all are connected through Gigabit Ethernet Switching network. In addition, it provides a wide variety of software tools, such as Rational Rose for Software Design, OPNET for Network Design, eTrust for Security Surveillance and Sybase/Oracle for database management.

**Interactive Multimedia and Virtual Reality Laboratory:** it provides the state of the art on teaching and research environment for multimedia systems, computer graphics, virtual-reality and advanced motion-detected computer interface studies. It’s equipped with an SGI Onyx 2 system, powerful graphics supercomputer, number of SGI workstations, cyber-gloves and motion tracking sensors to support the Department’s research into interactive media technology.

**Image Computing Laboratory:** it is funded by Innovation and Technology Commission, HK Jockey Club Charities Trust and Research Grants Council for pattern recognition, image analysis and synthesis multimedia application researches.
3D Motion Capture Laboratory: it is equipped with an optical motion capture system to support the real-time capture of 3D human motions for recording human movement in 3D environment.

Specialized Laboratory: it is a laboratory area for building up relatively long-term development environment for student and staff projects on the CS strategic research foci on software engineering, information security, mobile computing and multimedia computing.
ii. **Organizational Structure**

Generally, the strategic planning of department is mainly governed by the head and associate head of CS Dept and the management of daily operation is more likely in fat or single-level structure. Therefore, on the execution level of department, the responsibility of each staff will be accountable or regulated by some of 7 internal committees. My responsible functional group, **Web Team**, is a sub-division under Lab Management Committee.

![Organization Chart of CS Dept](image)

**Figure 1: Organization Chart of CS Dept**

**Departmental Executive Committee:**

- Formulate and keep under review policies, mission and objectives
- Determine and modify the internal management framework
Integrated People-Oriented Development Scheme

Research & Research Degree Committee:
- Coordinate, monitor progress and report the performance of research projects
- Evaluate the research degree applicants and do quality assessment

Academic Development Committee:
- Give advice on the development of new or existing programmes and courses
- Review study plan, time schedule, syllabus and external accreditation

Lab Management Committee:
- Prepare lab budget plan for Resources Committee
- Formulate the lab operation polices, guidelines and procedures
- Report the operational management on infrastructure and web applications

External Development Committee:
- Promote the departmental programmes and arrange publicity activities
- Liaison with alumni and visitors

Staffing Committee
- Manage the matters related to staff recruitment, re-appointment, promotion, crossing of efficiency bar and substantiations
- Review staff performance every year
Resources Committee:

- Arrange the resources on supporting teaching activities, labs and general office
- Plan the allocation of office and lab space

Each committee consists of one chairman and at least 3 committee members. Most CS Dept’s daily operations are guided by decisions drawn from corresponding committees. Given CS Lab as an example, the strategic plan and direction of CS Lab support is constructed by Departmental Executive Committee. Every year, the lab manager should consolidate the user requirements from Research & Research Degree Committee and Academic Development Committee to prepare lab equipment budget plan. With the approval by Lab Management Committee, the approved budget plan will be handed over to Resources Committee to allocate resources in terms of dollars or labors. Sometimes, CS Lab is also responsible to provide technical support for marketing promotion and external academic events held by External Development Committee. After the end of every academic year, the evaluation of CS Lab staff performance will be discussed as a case by case on the agenda of Staffing Committee.
iii. **Overview of IT Support**

Since most CS staffs have in-depth and wide-range of IT knowledge, the demanding of IT support is comparatively higher. For example, CS Dept most likely upgrades their server farms for every 2~3 years and core network devices for every 3~5 years. In addition, the technical staff should pay more attention on security, reliability, performance and scalability of every software application because both of CS students and staffs need higher service standards and system requirements for their teaching, learning and research support. For example, the multi-resilience of user home storage in CS Dept. If anyone loses a file, the system can help to restore the latest backup copy, which is backed up no longer than 4 hours, within few minutes. Another example is about one course to teach network security about how to breach the normal communication protocols and create network security problems. The technical staff should spend more time to prevent security holes and assume penetration trials by students to be safe without making troubles for others. Although it rises up the difficulty to manage the networking system, it also provides opportunity for CS technical staff to learn more or purchase sophisticated IT equipment to handle some scarcely advanced problems.

Besides the IT infrastructure, CS Dept has lot of software support tasks. For common client-based software, technicians will periodically upload the latest software on the PC workstations as student learning toolkits on LAN. For web-based applications, CS Dept formed a group of programmers named Web Team for web application development.
and teaching courseware support. The programmer who belongs to Web Team should demonstrate how to illustrate the analytical theories, mathematic algorithms, design patterns, implementation frameworks and different methodologies applied on Computer Science subjects. To be concise, the IT supporting tasks can separated into two parts and are responsible by two teams respectively as shown on Figure 2.

![Figure 2: Overviews of IT Support Management in CS Dept](image)

Besides, the Web Team members also develop some in-house systems to streamline the workflows of administrative operations. To improve the efficiency and productivity of CS Dept, there are 7 major in-house systems dedicated for specific purposes:

- **Final Year Project Management System (FYPMS)**
- **Placement Management System (PMS)**
- **Inventory Enquiry System (IES)**
- **Programming Assignment aSSessment System (PASS)**
- **Room Booking System (RBS)**
• **People Accommodation System (PAS)**

• **Research Student Information System (RSMS)**

From the management viewpoints, the Web Team has contributed some unique values for department. For example, the demonstration of best development practices based on classical software engineering methodologies. Moreover, they can closely customize the user requirements of the application specifications for CS Dept with higher user responses. Without the support of Web Team in CS Dept, it’s impossible to implement properly with any 3rd party application with tailor-made customization. On the contrary, it sometimes is difficult to solicit buy-in systems to reach the technical hurdles required by CS Dept.
iv. Present Situation and Business Objectives

Nearly 66% of lab resources are estimated to spend on daily operation and maintenance. The significant demand on lab resources triggers the management concerns about the cost of ownerships. Our approach is to evaluate whether any possible solution automate some low-tech tasks. For example, some tedious administrative tasks can be off-loaded to clerical staff by purchasing 3rd party system or plug-in appliances. Then, the technical team can release labor forces to focus on its original major duties or start some innovative projects which will be more beneficial to department in long-term.

On the other hand, the lab manager has stated a long-term mission for Web Team to integrate all in-house developed software into a single wrap-up interface as web-based dashboard control for each user. To sum up in conclusion on the later chapters, it depends on the pre-condition to standardize the role-based access rights across different web applications firstly developed.

Besides the above concerns of system dependence, the existing operation also highly relies on staff dependence. CS Lab assigns many risky administrative tasks to one Assistant Computer Officer (ACO). He mainly is responsible to maintain centralized system database for all CS Lab management. More than half of his working time is spent on database administration and user account management. The access right control for different web applications is distributed to be managed by several technical staffs or
administrative staffs. Since there is no clear guideline and code of practices to rule out the administrator how to correctly assign user access rights, CS Lab understands the centralized administrative tasks are very risky. Although the experienced ACO can help to resolve the conflicts among different user roles on different applications and also knows how to change the user roles with corresponding change of the working status, department still worries about business continuity when he takes long vacation leaves. Therefore, that’s why Web Team needs a standardized access right control framework to manage all of user roles across different web applications or administrative systems to avoid this preventable problem happened in CS Lab. Moreover, the number of system growth will be continued in the coming years. The standardized role-based access right control framework becomes more and more user demanding.

For every system development opportunity, the Web Team should evaluate whether it is feasible to become a project based on the user impacts and usage frequency. To improve the efficiency of technical staffs in CS Dept, Web Team has mission to automate all labor-intensive tasks by either self-developed or buy-in systems. The administrative procedures should be standardized into a framework to be easily followed by technicians or clerical staffs. There are some scenarios that it is advantage for Web Team to develop a new role-based access management system. For examples, it reduces the coming development resources including time, cost and labors to improve the efficiency and effectiveness of user account management. Besides, there are several
frequently asked questions from different stakeholders would be expected to be answered:

1. Department Head: “Can CS Dept provide clear user access right management guidelines for each system?”

2. Academic Staff: “May I know all the user access rights of his/her supervised research staff on different systems?”

3. Student: “May I know his/her own user access rights on different systems?”

4. Graduated Student: “Can I make sure that all access rights are removed after graduated?”

5. Administrator: “Some systems’ administrative rights are distributed, can I trace it all changes?”

6. Visitor: “I only stay in your department for one week. Why the administrator can’t assign my wanted access rights correctly on time?”

7. Clerical Staff: “About user management, I have no overall picture or concept. I just do what I have been instructed. There is no guideline to care about the conflict of wrongly user role assignment.”

8. Programmer: “Is there any standard to follow on role assignment? Otherwise, I still need to spend more time to maintain different approaches of them.”

9. College Dean: “As the highest IT technological level in our College, can CS Dept share your experience or provide any recommendation for access right control management to follow as the best practice guide?”
To sum up, there are 5 main business objectives, as shown on Figure 3, to be considered as evaluation criteria to measure the efficiency and productivity of any recommended software development proposal. The highest total rating means it is the most effective solution to provide benefits for the stakeholders of CS Dept.

Figure 3: The business objectives of software development projects in CS Dept
Part II: Case Analysis

Chapter 1: Problem Identification

In CityU CS department, most of administrative works can be performed by in-house developed web applications. The login function of the web applications is standardized by the integration of Single Sign-On (SSO) server to centralize all user account management. On Figure 4, SSO only streamlines the processes of user identification and authentication which are the only part of whole access control management concept. There is no common authorization framework to grant the appropriate multiple user access right(s) on each web application by the user owned role(s) for each user.

Identification & Authentication

Authorization

Different Web Applications

Same Web Application but Different Rights

Figure 4: The Workflow of Web Application Login
For every new coming staff or student, user account administrator should assign the appropriate access rights on required systems. The granted access rights provide essential and sufficient CS Lab services to perform functions under specific user role. These access rights sometimes would be adjusted on different stage of their working period. For example, CS Dept has guidelines to restrict user not to allow for lending any equipment from inventory system within 2 weeks before contract end date.

![Diagram showing RBAC methods for CS Web Applications](image)

**Figure 5: 3 RBAC Methods for CS Web Applications**

In CS Dept, there are 3 different Role-Based Access Control (RBAC) methods customized on different systems. Generally, each user has his/her unique CS user account and owns his/her personal details in the SSO server and CS Management Information System (CSMIS) database server respectively. Refer to the operational flows diagram as shown
on Figure 5, the 3 RBAC management types can be shown to have slightly different from each other.

- **Type 1**: The RBAC information of web application is directly relied on the information provided by SSO and CSMIS server.
- **Type 2**: The RBAC information of web application is relied on the application linked internal database, which is kept in synchronization with some relevant personal information from CSMIS server.
- **Type 3**: The RBAC information of web application is stored into a proprietary database, in which the administrator should assign the corresponding roles or rights directly for the application.

Because of historical reasons, few legacy web applications must be specially configured in hybrid mode, which means to mix-up with any 2 of above types for the management of RBAC information. At the present situation in CS Dept, only one administrator, who is dedicated for user account management, is able to have sufficient experience enough to make correct judgments on the assignments of user access rights over different web applications consistently.
1.1. Problem Statement

In view of the inefficiency and unproductive administrative procedures related to role assignments across different web applications in CS Dept, Web Team should improve the user satisfaction by proposing one recommended solution to CS Lab Committee with the following requirements:

- Study the existing problems about user access right management controls across different web applications.
- Evaluate different approaches with alternative recommendations.
- Justify this project to be highest priority for the improvements on the efficiency and productivity of CS Dept better than other alternatives.
- Propose a framework to standardize the user access control management of individual users across different web applications with scheduling.
1.2. Project Mission

To improve the administrative efficiency by centralizing the present loosely distributed RBAC management, a development scheme including one proposed system is to standardize the access control management practice and to integrate all related role-based information as a guided structural framework for web application developers to follow.

The project name is “Integrated People-Oriented Development Scheme” coded as (IPODS) which define a people-oriented generic centralized database schema to integrate with all user/group access rights on each web application. It also aims to simplify the authorization process by centralized administration.

Besides the above concerns of administrative efficiency, it helps to prevent data inconsistency by distributed assignments and conflicts of access rights across different web applications. In addition, it provides the web application control assignment logging for security audit trail and schedules the access rights on different working time period. Furthermore, it implicitly enforces the documentation of access rights for web applications to avoid any strictly administrative guidelines which sometimes are wrongly interpreted by different administrators. Finally, the concept of “service profiler” also can easily present the access right information for application users, system administrators and software developers. It also supports Service Oriented Architecture (SOA)
framework to compile the modern standard of business needs. The draft of design overviews is shown on Figure 6.

![Diagram of Integrated People-Oriented Development Scheme]

**Figure 6: The Overview of Integrated People-Oriented Development Scheme**
Chapter 2: Literature Reviews

There are many ways to implement a new integrated system to consolidate the access control and authorization rights across heterogeneous backend infrastructures (Chen, Chen, Chu, & Wang, 2006). From the viewpoints of management, our implementation should balance the total cost of ownerships and technical advancement on system design. After reviewing technological trends, the following 3 approaches are summarized to be recommended as the structural framework design of IPODS.

2.1. Service Oriented Architecture (SOA)

In order to gain the benefits of SOA (Raghavendra, 2007), the modern system design should have the following generalized characteristics:

- To separate the authorization feature into a distinct unit, which is accessible over a network in order to be combined or reused in the production of business applications.
- To make the authorization function to become a loose coupling service.
- To provide platform to realize rapid system changes with lower cost implications and higher quality improvement.

From the source of parasoft.com, SOA enables IT departments to make the transition from an application-centric view to process-centric view. IT departments should provide
flexibility to combine some common business services from multiple applications while delivering end-to-end service supports for business processes in consistency. Since the structural design enables loosely coupled integration, IT departments can upgrade or change applications transparent to users without impacting other applications and supporting parties.

In addition, one web blog (Harris, 2008) provides good summary that “SOA is the architectural style that supports loosely coupled services to enable business flexibility in an interoperable, technology-agnostic manner. SOA consists of a composite set of business-aligned services that support a flexible and dynamically re-configurable end-to-end business processes realization using interface-based service descriptions”.

To sum up, SOA is concluded to provide benefits in 4 basic categories:

- Reducing integration expense
- Increasing asset reuse
- Increasing business agility
- Reduction of business risk
2.2. Role-based Access Control (RBAC)

To take advantages of RBAC (Guerin & Lord, 2003), the new authentication and authorization workflow design will have the following characteristics:

- To reduce complexity and cost of security administration.
- To align with organizational structure.
- To improve the productivity of developers with generic access control information model.
- To simplify the regulatory compliance and system administration.
- To enhance systems security, integrity and manageability.
- To reduce the number of relationships between access right assignments and user objects.
- To reduce the number of changes to access control information because role-based organizational activities can be considered to be more stable and are rarely changed.
- To improve the accuracy of access control information.

RBAC is one of non-discretionary (or mandatory) access control techniques. Discretionary access control is an access policy determined by the owner who decides what privileges the users have. On the contrary, mandatory access control is an access policy determined by the system rather than the owner. It also applies on the multilevel systems that process highly sensitive data. There are 3 primary rules defined for RBAC (Zhang, Zhang, & Ravi, 2006):
• Role Assignment: A subject can execute a transaction only if the subject has been assigned a role.

• Role Authorization: The role as a rule to restrict the authorized subject to have transaction executing permission.

• Session Authorization: A subject can execute a transaction only if the transaction is authorized for the subject’s active role.

On the other hand, RBAC has 4 sub-categories: RBAC0, RBAC1, RBAC2 and RBAC3 (Sandhu, 1996).

• RBAC0 is the base model or named as core RBAC. A role is a functional or organizational job description and each user can be assigned with many roles.

• RBAC1 is the hierarchical RBAC to add the concept of role hierarchies. Role hierarchies define inheritance relations among the roles in terms of permissions and user assignments.
• RBAC2 only enforce more strong restrictions by determination of acceptance with a collection of constraints.

• Finally, RBAC3 inherits all the characteristics from both of RBAC1 and RBAC2.

Refer to the white papers of Sun Microsystems, RBAC provides the following characteristics and benefits as shown on Table 1:

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Benefits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Streamlined access control</td>
<td>• Simplifies assignment and management of user access</td>
</tr>
<tr>
<td>processes</td>
<td>• Creates and manages roles rather than users</td>
</tr>
<tr>
<td></td>
<td>• Achieve compliance via access certifications and “Separation of Duties” policy enforcement</td>
</tr>
<tr>
<td></td>
<td>• Aligns Business and IT with a common vocabulary for IT access permissions</td>
</tr>
</tbody>
</table>
| Enhanced audit effectiveness | • Ensures ongoing understanding of access
• Reduced risk of security violations or access control related deficiency – improper access will occur and that enterprise security policies will be violated |
| Robust Role Management | • Robust life cycle management which includes versioning, consolidation, history and ownership
• Enhanced workflow engine to manage lifecycle of roles
• Complete setup of security, workflow and auditing features to manage the lifecycle of rules |

Furthermore, the latest analysis (Piromruen & Joshi, 2005) of RBAC mentions that time constrained information sharing between different systems is becoming a common phenomenon. A flexible and efficient mechanism is needed to support short term time-based sharing policies between transient partners. In particular, the interacting domains need to establish a time-based inter-domain access policy without violating the original time-based security policies of the individual systems. Therefore, the new design of RBAC Framework should include **time constrained interoperation** across multi-domain environment.
2.3. *eXtensible Access Control Markup Language (XACML)*

Refer to the IBM Development XML Library, XACML is the XML document standard widely used for access control purposes. It is composed by many components described on Figure 9.

![Diagram of XACML Implementation Design]

**Policy Enforcement Point (PEP)** is to handle the landing of authorization request. It creates and sends an XACML request to the **Policy Decision Point (PDP)**, which evaluates the request and sends back a response. After evaluating the relevant policies and the rules within them, PDP will draw into a decision. There are a number of policies and policy sets store in **Policy Access Point (PAP)**. The PDP does not evaluate all of them but only the relevant ones are chosen for evaluation, based on the policy target, which contains information about the subject, the action, and other environmental properties.
Afterwards, the PDP may also invoke Policy Information Point (PIP) service to retrieve the attribute values related to the subject, the resource, or the environment. The authorization decision arrived at the PDP is sent to the PEP. To fulfill the obligations and authorization decision validated by PDP, PEP response either permits or denies access.

According to the in-depth analysis of XACML from (Kay, 2003), the followings are the summarized advantages of XACML over other access-control policy languages:

- **Rule Reusability**: Security Administrators can describe an access-control policy once, without having to rewrite it numerous times in different application-specific languages.

- **Code Reusability**: Application developers don’t have to invent their own languages and write code to support them because they can reuse existing code with standard format.

- **Standard Format**: XACML is intended to be primarily a machine-generated language. XACML creators expect that easy-to-use tools for writing and managing XACML policies will be developed because they can be used with many applications.

- **Consistency**: A single XACML can be applied to many resources. This helps avoid inconsistencies and eliminates duplication of effort in creating policies for different resources.

- **Scalability**: In a large organization, a policy for a specific site might refer to both a companywide policy and a country-specific policy.
Chapter 3: Managerial Justifications

3.1. The Trend of Integration Approach

Before 2003, each system in CS Dept was developed separately and operated standalone without data inter-exchange with other systems. At that period, most of systems were firstly grounded up from CS undergraduate final year projects. Afterwards, CS programmers always pick up the post-deployment enhancement and further maintenance works. During this time period, the quality of final outcome was unpredictable because the past CS Lab Staff had insufficient knowledge about the original design concepts. In addition, CS Lab had no labor resources to define any best practices, any coding standards or design methodologies to guide the system developers.
In 2003-2005, CS Lab consolidated different production database servers into single one, named as “CS Management Information System” (CSMIS), and also integrated identification and authentication process across UNIX, Linux and NT OS platforms by Single Sign-On (SSO) Service with Light-weighted Directory Access Protocol (LDAP). These two enhancements of the system infrastructure provided better application development environment. With sharing the same coding practices, the developers modified the past web applications with standard authentication web module and shared database retrieval to achieve higher performance on coding efficiency. On the other hand, the administrator centralized the distributed data administrative rights to prevent inconsistent management by different application administrators.
From 2006 to now, CS Dept has formed a new task force group called Web Team which focuses on the web application development by applying the best practice of software engineering skills. After more than half year of evaluation and research studies and good leadership by CS Lab Committee, Web Team has owned its knowledge on project management, web application technologies, reusable design patterns, coding guidelines and quality standards with international compliance. Refer to the section about the present situation of IT Infrastructure in CS Dept, lots of in-house applications were rapidly developed and rolled out. It led the workload of our administrator increased sharply. Therefore, the second level of integration about authorization framework should be implemented by the standardization of the user access rights in generic model like the proposed solution, IPODS.
In the coming future (planned to be starting from 2010), Web Team will propose to further integrate all the existing web applications running on a single management console, named as CS Information Management Console (**CSIMS**), which also be developed on top of IPODS platform. Besides large scale integration planning, there are 2 administrative workflow automation projects, like Postgraduate Student Admission System (**PSAS**) and Course Administration System (**CAS**). Since the labor resources of Web Team is insufficient to implement all the projects at the same time, the next section about the project priority assessment would help the senior management to select the right choice which is the most beneficial to CS Dept.
3.2. Project Priority Assessment

<table>
<thead>
<tr>
<th>Table 2: Project Priority Assessment Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item</td>
</tr>
<tr>
<td>Project Name</td>
</tr>
<tr>
<td>System Code</td>
</tr>
<tr>
<td>Description</td>
</tr>
<tr>
<td>Usage Frequency (+ve)</td>
</tr>
<tr>
<td>User Impact (+ve)</td>
</tr>
<tr>
<td>Dependence (-ve)</td>
</tr>
<tr>
<td>Cost of Ownership (-ve)</td>
</tr>
<tr>
<td>Expected User Demand Growth (+ve)</td>
</tr>
<tr>
<td>Overall Ratings</td>
</tr>
<tr>
<td>Managerial Priority</td>
</tr>
</tbody>
</table>

Refer to the project priority assessment table on Table 2, it evaluates the project priority by the summation of the proportion of rating values between 1 to 10 on five assessment criteria such as usage frequency, user impact, dependence, cost of ownership and expected demand growth, which all have already been mentioned on the section iv of Part I. In conclusion, by comparing the calculated overall ratings, the sequence of project priority is IPODS, CSIMC, PSAS and CAS in descending order.
For long-term planning, Web Team should provide a single console-like web application to wrap up all other web applications into the same working environment on one web browser. From this user viewpoint, it provides the most convenience and improves the highest working efficiency for users to interchange the application data between different web applications. For the success of this high level integration, there are two major pre-conditions. The first challenge is how to apply “Rich Internet Application” client-side technology to wrap up all web applications running normally under a web browser. The second problem is how to generalize all the access control rules across different web applications. Since both of the problems generate demands on the standardization of RBAC model for authorization, the system dependence and user impacts led IPODS become the critical point to make the continuous system growth which assume to make benefits for CS Dept in long-term.
Chapter 4: Analysis and Findings

At the present situation, it is very difficult to collect the real data by system logging under the new enforced personal data (privacy) ordinance compliance under university policy. However, a simulation model by Microsoft Excel spreadsheet helps to imitate the real workload of user access right control operation. The behavior of simulation can be adjusted by parameters and also inherits the characteristics of randomness.

Firstly, the batch mode of user account changes on each term is excluded for this analysis because the efficiency of our automation scripting is higher enough to make time spent on role assignment for each user not significant. Secondly, it doesn’t count the time involved on the role re-engineering by optimization. Thirdly, except the percentage of human mistakes, I assumed all other decision variables are randomly generated and spread in normal distribution. Through the comparison with number of observations, the heuristic quantitative simulation model can be finalized by the reasonable adjustment of its parameters.

Practically, CS Dept needs to add or change 2 user accounts daily. Each user account may involve 5 different systems on average. If nothing wrong, each role assignment will be finished on around 5 minutes. Otherwise, if some human mistakes (e.g. administrative staff forgets to activate the user account or assigns the inappropriate
access control rights to the user.) or controllable external factors (e.g. duplicated roles and role conflicts.) are happened, this task assignment will be added with penalty time up to 30 minutes. The standard deviations of parameters are estimated by reasonable judgment on the value range of random distribution.

Refer to the operational workflows between the present situation and the IPODS solution on Figure 13, the processes with blue colored words include human mistakes and the processes with red colored words are assumed by manual procedures to recover the mistakes.
Since there is no historical statistics about the percentage of penalty role assignment suffered from all unexpected factors, I proposed two cases 1% and 5% error rate for comparing the differences of existing situation and the new proposed solution with IPODS. As shown on Appendix A & B, the 2 simulations are calculated separately on the same Microsoft Excel spreadsheet with 2 different error rates respectively.
Table 3: Forecasted performance gain by simulation

<table>
<thead>
<tr>
<th>Case</th>
<th>Error Rate</th>
<th>Estimated Time at Present Situation</th>
<th>Estimated Time with IPODS</th>
<th>% of performance improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1%</td>
<td>52.34618698</td>
<td>10.55414808</td>
<td>496%</td>
</tr>
<tr>
<td>2</td>
<td>5%</td>
<td>62.57895069</td>
<td>12.40997555</td>
<td>504%</td>
</tr>
</tbody>
</table>

From the simulation summary on Table 3, the increase of error rate from 1% to 5% leads around 20% downgrade of overall task performance. At the same time, the expected performance gain of IPODS is nearly 500% by this minor change of the operation workflows between two situations. With IPODS, the time required to finish the role assignment is around from 10 to 12 minutes. On the contrary, CS administrator always spend nearly 1 hour to handle the same kind of ad-hoc change request on role assignment at present.

Although the assumptions cannot be validated by past user statistics, the transformation of workflows can be verified by “Walk-through Audit” approach to prove the final result whether the parameters ($\mu$, $\varepsilon$, $\rho$ and $c$) of simulation model on Figure 13 will be matched to the expected values after implementation.
Chapter 5: Design & Implementation

After case studying on the Part I, CS Dept are looking for a solution to achieve better access control management with less user and resource impacts. The Web Team also place importance on increasing operational productivity, reducing maintenance support efforts, ensuring compliance of system audit and raising the quality of user service level. With full justification and management approval, Web Team can start to design a new operational model about user account management with more proactive response features. The IPODS design initiates paradigm shift to higher service agreement with flexible user profilers rather than the traditional user account management with basic functional permission assignments by referring to the Access Control Management Model shown on Figure 14.
5.1. Conceptual Design

According to the diagram shown on Figure 15, the whole design not only can perform the general RBAC management but also handle the more complex system architecture. On the left hand side of user assignment, different user groups (e.g. academic staff, clerical staff, technical support staff, student, etc.) can be consolidated by similar role assignments for their members. By the other means, they have the same roles according to their organizational positions or their functional requirements. On the right hand side of permission assignment, the system functional roles (e.g. assessor, operator, database administrator, etc.) also aggregate the similar permission control assignments across different systems. From the design specification, the differences between existing web application systems should be studied for the design of generic data structure on role assignments. Otherwise, the RBAC matrix only provides RBAC information but can’t synchronize the actual permission control rights to the present heterogeneous systems for execution.
Refer to Figure 16, the RBAC session control is managed by Single Sign-On (SSO) system through the authentication process by centralized LDAP people user directory service. The overall system flow design follows the generic RBAC1 model to construct role hierarchy for role inheritance with reflexive, transitive and anti-symmetric relationship’s characteristics. However, in the design of IPODS, RBAC1 will be enhanced by a timer control to activate the permission assignment at specific time period. So, IPODS can assure that the user access rights are properly granted in the period of their valid working time only. Although RBAC2 model has role constraints checking on sessions and permission assignments, it increases the complexity of IPODS design and also cannot break the whole RBAC system down into two processes by authentication (in LDAP) and authorization (in IPODS) separately.
5.2. Implementation of XACML Architecture

XACML stands for XML based access control language, which is designed for simple syntax, strong expressivity and machine readable. To compile with the Organization for the Advancement of Structured Information Standards (OASIS), it is widely adopted both in industry and academia. The source code of XACML can be downloaded from open source group SourceForge.net or Sun Microsystems Laboratory. Besides, OASIS defined the architecture and the components for the creation, enforcement and management of access control policies written in XACML (Staggs, 2004).

The implementation simplifies the full feature set design of OASIS XACML architecture shown in Figure 9 and maintains the 4 core components used in this project:

- Policy Enforcement Point (PEP) – intercepts the request for authorization with all available information about the user, the action being requested, the resources and the environment
- Policy Decision Point (PDP) – based on the request for authorization received, retrieves the appropriate authorization policy from the Policy Administration Point(s)
- Policy Administration Point (PAP) – manage authorization policies by authorized person(s) appointed by the application managers
- Policy Information Point (PIP) – source of information at the disposal of the PDP in order to evaluate authorization policies

![Diagram](image)

**Figure 17: Dataflow Summary of XACML**

In brief, a user request for permission generates a “decision request” by PEP. The request statement describes the subject making the request and the resource wanted to be accessed. Then, it passed to PDP for decision making with the policy database from PAP. Finally, it creates a rule in PIP to be used for accessing information over the application environment. Actually, it was more complicated for this authorization processing. I simplified the overall flow into 5 major steps as shown on Figure 17.
5.3. Project Plan

Since this is a new project of CS Dept in parallel run with MBA project, the first fourth periods shown on Table 4 are aligning with what I had committed on every project group meeting before.

<table>
<thead>
<tr>
<th>Period</th>
<th>Task Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sept – Oct, 2008</td>
<td>Background Researches and Alternatives Evaluation</td>
</tr>
<tr>
<td>Nov, 2008</td>
<td>Proposal Drafting and Supervisor’s Review</td>
</tr>
<tr>
<td>Dec, 2008 – Feb, 2009</td>
<td>Feasibility Studies and Managerial Justification</td>
</tr>
<tr>
<td>Mar – Apr, 2009</td>
<td>System Specification Design</td>
</tr>
<tr>
<td>May – Dec, 2009*</td>
<td>Application Architectural Design and Implementation</td>
</tr>
<tr>
<td>Jan – Sept, 2010*</td>
<td>Beta Testing, User Acceptance Test and Bug Fixing</td>
</tr>
<tr>
<td>Oct – Dec, 2010*</td>
<td>Pilot Test and User Reviews</td>
</tr>
<tr>
<td>Jan, 2011*</td>
<td>Official System Launch</td>
</tr>
</tbody>
</table>

Remark (*): The proposed implementation phase would be changed depending on web team workload and senior management decisions.

As the same as other academic departments, CS Dept only allow the critical IT system changes in the non-peak time period. In the time period within semesters, all technical staffs should be concentrated on their daily administrative operations rather than new development tasks. Therefore, the beginning of project implementation only can be allowed to start from the May of this year because the less user impacts are expected in the end of semester B. In addition, it possibly provides enough buffer time for project implementation and system testing which maybe accidentally lasted for the whole summer semester.
To review and evaluate the result after pilot testing, the good timing is in the beginning of new academic year between October and November which always is the highest system utilization in the year. Since the operation of IPODS is transparent to most of client users but the system launch involving changes over the past authorization methods in cut-through way. The system migration is appropriate to be scheduled on the break time of Semester A.
5.4. Technical Specifications

IPODS will be 100% in-house developed system by Web Team. There is no need to purchase any new equipment or software specified for this project. All items mentioned on below Table 5 are existing servers by common system provisioning in CS Lab. The additional system computation loads and storage spaces are not significant under the flexibility of virtualized infrastructures in CS Dept.

<table>
<thead>
<tr>
<th>Infrastructure Components</th>
<th>Platforms</th>
<th>Justifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
<td>Sun Solaris Operating System</td>
<td>Critical system needs to be executed on reliable OS.</td>
</tr>
<tr>
<td>Directory Servers</td>
<td>Sun Java System Directory Server</td>
<td>Enterprise level and open-platform architecture to consider the scalability and compatibility in future.</td>
</tr>
<tr>
<td>Database Servers</td>
<td>Microsoft SQL Server</td>
<td>Standard consolidated database system used in CS Dept.</td>
</tr>
<tr>
<td>HTTP Servers</td>
<td>Apache HTTP Server</td>
<td>Our team has long-term shared experience on this open-source and provides flexibility to integrate with other server-side scripting language.</td>
</tr>
</tbody>
</table>
5.5. Project Team

In CS Dept, Web team is a specific project team to be responsible to pick up any web application development task. It consists of 4 members, which are database administrator, two application programmers and one team leader. Normally, the software design workload should be shared around each others. It’s not only get involvement with all members but also try to look into depth from different dimensions and knowledge specialists. Since each team member also take additional support roles or teaching tasks daily, department head assumed that they should consider the application development projects in lower priority and no one would be dedicated their tasks on software development only. To resolve the above mentioned problem about insufficient labor resources, 3~5 student helpers are hired as part-time mode to work with web team together. Although some student helpers had done pretty well on coding, this project still has lot of security issues and confidentiality concerns. As a Web Team leader, I should pay additional efforts to double-check their program codes and also don’t let them to get in touch of any confidential data. Finally, the implementation of the technology of XACML and Role-Based Access Control (Version 1) also is a new big challenge to web team because of lack of similar experience. It is expected to spend more time on the feasibility studies of system impacts.
5.6. Project Lifecycle

The project workflow shown on Figure 18 is a standardized workflow design for software development project in CS Dept. It always is used as a benchmark to guide the working progress of each project. To consider the project complexity of IPODS, it is difficult to start implementation without any clear definitions about user requirements and technical concerns. There are 2 phases which are difficult to be time scheduled. The first one is “Requirement & Design Plan” that maybe continued in 2010 because it is time-consuming process to prepare software design specification for approval. Another one is “Development & Testing” that is predicted to be a great challenging task for Web Team and need to collect user feedbacks from different stakeholders.

![Image of the workflow diagram showing the Project Lifecycle in CS Dept.](image)
5.7. Implementation Methodologies

For each CS development projects, the bottom-up and top-down requirements should be carefully investigated and also be reviewed time by time. Generally, the Web Team periodically reviews the following 3 external factors whether it will be justified for project continuation.

- **The Strategy of CS Lab Committee** – The project scope should be tangible, visible and related closely to department goals. The benefits of projects must be evaluated by cost saving, effectiveness or performance.

- **The Satisfaction of System Users** – The program purposes should be easily understood and the workflows are clearly streamlined. Before system rolling-out, user survey and performance optimization must be finished to achieve higher customer satisfaction.

- **The Dynamic Changes of Latest Technology** – As the department of Computer Science, the system should be compiled with international standards, good software engineering practices and scalable design frameworks.
On the other hand, the IPODS was expected to be a complex and large-scale system with lot of interfaces to other system. Although most of past projects are developed by prototyping in very short period of time, the system design specification of IPODS should be clearly defined before starting implementation. By the other means, we always started to build prototype at first and then revised it by numbers of iterations in the past. In this project, we should seriously evaluate the details of system design and then circulate the design document for comments in CS Lab Committee. The formal design may apply the techniques of Unified Modeling Language (UML) to define data schema, data flows and entity behaviors. After finalizing the design by the committee members especially counter-checking or endorsement by system administrative staff, the specification will be approved for implementation because we should clarify all the unknown impacts and workflows understood by all participants.
5.8. Testing Approaches

In order to ensure the reliability of IPODS under critical environment, number of tests will be exercised for validating the correctness of functions and data flows. To sum up, testing will be performed by 2 approaches: walk-through and simulation. In a walk-through test, the programmers will follow the logics previously defined by UML and check up the executing results or processing behaviors whether are matched with expected outputs should be. In simulation, a number of stress test cases will be prepared to try running every UI features and system functions. To compare the final results with expected outcomes, it can validate whether the system design and coding is correct or not.

Besides formal testing exercise, the good practices of software engineering design provides automatic self functional testing by embedding with unit test functions in each module of IPODS. On the other hand, it can be considered as one of the other critical web service, like DNS and DHCP, in CS Dept. The CS network monitoring appliance not only check the availability at network and system layer but also periodically send the request to validate the application layer of IPODS especially during early deployment stage.
5.9. Deployment & Continuity Planning

Since IPODS is to integrate with more than one other web application systems, it is impossible to be in parallel run with old systems because the duplication of user data rise higher security risk than its system fails. On the other hand, cut-over conversion is the most inexpensive method but also is the highest risk method because the system can’t be rolled back (Oz, 2006). With the limitation of IPODS design and functionality, it can’t be broken down into different modules for phased migration. Therefore, the migration will be introduced by pilot conversion. It means that one web application system will be firstly chosen to integrate with newly developed IPODS for test run in a short period of time. If having any problem, it only affects the trial one application system and prevents the problem spreading out to other systems over the whole department. Pilot conversion reduces risks and increases trust to new system users with trial experience for reference.

Since IPODS will be considered as new critical service in CS Dept, the impacts of business continuity becomes hot topic to be discussed internally. To prevent any inconvenience or unpredictable disaster caused by IPODS deployment, CS Dept should reserve sufficient spare hardware resources which are enough to replicate the whole set of infrastructure for pilot conversion. To take the advantages of server virtualization in CS Lab, all the critical systems can be recovered within an hour in case of any service interruption.
5.10. Project Risk Assessments

As mentioned on previous section, IPODS is transparent to end users so that the switching cost only is the service down time for deployment. The high risk tasks, like deployment or system upgrade, are always scheduled at the weekend in non-peak season, for example, one Sunday in the semester break. Comparatively, the impacts on administrators are higher than users because IPODS centralizes the access control management. For commercial companies, it leads job security concerns and emotional resistances due to service consolidation. Since the technical support of CS Dept is not a big team, every administrative task is under controlled by CS Lab. If the outcome benefits are not over-weighted, all administrative staff and technical staffs are welcome to shift from the present annoying working situation to more productive and maintainable new environment. The following table will show the possible business risk led by switching barriers.

<table>
<thead>
<tr>
<th>Type of Switching Barriers</th>
<th>Probability (1:Low – 9:High)</th>
<th>User Impacts (1:Low – 9: High)</th>
<th>Risk Rankings (1:High – 5: Low)</th>
</tr>
</thead>
<tbody>
<tr>
<td>i. Labor Cost</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>ii. Equipment Cost</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>iii. Emotional Resistance</td>
<td>5</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>iv. Security Concerns</td>
<td>5</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>v. Insufficient Knowledge</td>
<td>4</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>vi. Startup Change</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>vii. Maintenance Cost</td>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>vii. Exit Change</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>
Chapter 6: Evaluation of Implementation

Since the IPODS project is still under the phase of planning and designing on May, 2009, the whole project is difficult to be evaluated at that moment. However, I summarize the expected key benefits as the future benchmarks for evaluation.

According to the previous mentioned objectives, IPODS could project into the following benefits:

- Increase internal efficiency and productivity of CS Dept
- Increase the coding efficiency of application programmers in CS Lab
- Improve fast integration or consolidation by system administrators in CS Lab
- Enhance higher compliance of user requirements and service level in CS Dept
- Support ongoing operation and strategic IT planning in CS Lab
- Generate new user demand and opportunity for system growth

Normally, the effectiveness of Information Systems are defined by quality metrics like service quality, information quality and user satisfaction, ... etc (Grover, Purvis, & Coffey, 2005). To evaluate the success of IPODS objectively, the numerical measurable metrics for the effectiveness of a new system should be applied as follows:
- **Mean Time to Repair**: How long it takes to resolve an outage to measure how easy of the system to be fixed.

- **First Fix Rate**: The percentage of incidents that are correctly repaired at the first time to measure the average system recovery time.

- **Change Success Rate**: % of changes or number of scopes that are successfully deployed without creating an incident to measure the flexibility of requirement changes on the system.

- **Server to System Administration Ratio**: Number of servers that can be handled by an administrator to measure the total cost of ownerships. (e.g. high performers can handle about 20 servers while the low performers can handle around 5 servers.)

In order to transform quality metrics into measurable domains, the quality of system can be evaluated by scoring scale with broader dimensions (Mani, Shenoy, & Dayasindhu, 2005):

- **Overall user-satisfaction level**

- **Application Availability** (e.g. Less than 0.1% to above 1.0% downtime)

- **Application Response Time** (e.g. Account Creation Process completed within 10 minutes, modification process within 20 minutes and deletion process within 5 minutes, etc.)
• **Estimated % of Project Scope Completion** – Compared with the defined features in the system design specification, the reasonable implementation successful rate should be higher than 90%.

• **Estimated % of Critical Contribution** to CS Dept – To determine how much importance of the IPODS in CS Dept. It can be measured by network proportion of its related Internet/network sessions daily.

• **Level of Security Compliance** – To measure the number of security incident or noncompliance reports through monitoring.

• **Significance Level of Data Restoration** – To understand how much impact for system data rolling-back if any incident was happened.

All the above metrics will be applied to measure different dimensions on the system performance and effectiveness of IPODS both on user service level and system functional level. It effectively provides the whole picture to measure IT efficiencies for non-profitable organization like CS Dept in CityU. Moreover, it clearly establishes accountability for desired outcomes in order to motivate project team members to achieve higher levels of performance. Finally, it provides valuable assessment vehicle for addressing concerns about the expectations of project stakeholders.
Chapter 7: Prospects for Further Improvement

If the IPODS platform is mature enough to become an open platform for centralizing the access control management in CS Dept, the following suggestions would be considered as feature extensions in future.

7.1. Separation of Duties (SoD)

With increasing the complexity of departmental processes and applications, administrators can no longer effectively manage SoD by manually check-ups, especially when the ability to prove SoD enforcement as a key audit concern or regulatory compliance (Fox Technologies, 2008). Therefore, the pre-defined rule sets of common SoD in CS Dept should be added into the role assignment matrix of IPODS in order to greatly simplify the consideration of security constraints during role authorization through the system.

7.2. Role Optimization

Role optimization originally is a sub-function of role engineering in the process of designing an RBAC system (Vaidya, Atluri, Guo, & Adam, 2008). A common approach of role optimization is to apply data mining techniques to maximize the searching efficiency of RBAC system from original permission assignment data. Possibly, the system can implement some simple weighting algorithms to analyze the existing
efficiency of role assignments. For example, there are less than 10 different common user roles in CS Dept. When the number of systems increases along few years, it generates more than 100 different roles but most of them are so similar. Besides some roles are too specific for niche purposes, other temporary roles for testing purposes are created once but then never been used again. The data analysis on role assignment data helps to screen out the current situation and provide just-in-time advice to decide whether to do role optimization.

7.3. **Automatic Account Replication**

To reduce the disaster caused by the single point of failure, IPODS only centralizes all the RBAC data and only perform the function of authorization. Actually, each system can survive its functions standalone without IPODS if necessary. It implies that each system should have its own user account database for service continuity. For example, similar to Google, Gmail and Picasa also have their own accounts can work independently. You only create an account on Gmail and then you just activate other system through its internal automatic account creation process. In IPODS, we can develop different scripts running on cross application system environment to perform the same function of auto-account creation.
7.4. XML Account Portfolio

XML Account Portfolio will show up the detail application services granted by IPODS in XML data format. It is developed as an extension of XACML to consolidate all the roles and access control rights of a person into a single file data format for the ease of exchange across different applications. Practically, it is not necessary to consider the portability of user right profiles because there is no international standard format at present and the XML account portfolio data is proprietary to IPODS for internal use only. However, we should consider the fast dynamic changes of IT technology. It provides data flexibility to choose different tools, platforms or even database servers. To achieve the benefits of data independence within CS Dept, XML provides common data exchange standard for data transformation.
Chapter 8: Summary and Conclusions

Although CS Dept has consolidated the user identification and authentication processes by SSO server in centralized approach, the authorization controls are still flexible and distributed across different application systems. There are 3 different types of RBAC management scheme to take the responsibility of authorization control in CS Dept. However, the operational complexity leads serious concerns about the limitation of system growth and gives heavy burdens to system administrators. In order to resolve the problems related to inefficient system design and unproductive administration, this report proposes an open system integration design scheme named “Integrated People-Oriented Development Scheme” and coded as IPODS, to integrate the user access control by a matrix of RABC framework under centralized authorization management policy. To consider higher service requirement and take the favor of cross application environment, its system structural design will follow service-oriented architecture (SOA) and its data will be in XACML format.

Back to the history of system development changes in the past 2 decades, the management of administrative systems was starting from distributed standalone style to be more centralized integration approach. To align with the IT business objectives of CS Dept, IPODS is justified by the highest overall ratings over the 5 measurements of priority assessment, such as usage frequency, user impact, project dependence, cost of ownership and expected demand growth.
With the implementation of IPODS, CS Dept will achieve the streamlined, robust and auditable access control management system to satisfy better integration and higher service level requirements. Through the simulation analysis, IPODS applied the tactics of updating account information in batch mode which helps to improve daily system administrative performance by 5 times. For example, the daily operational time for account management normally needs an hour to complete all required administration procedures but the IPODS will help administrators to complete their tasks within 12 minutes.

Due to the time limitation of MBA project period, it’s difficult to evaluate the project at the design and planning phase. However, based on the defined objectives and expected benefits, the intangible quality outcomes can be evaluated by some measureable IT effectiveness metrics (e.g. Mean Time to Repair) and subjective scale-scoring quality metrics (e.g. User Satisfaction Level).

The IPODS design brings out the important concept of platform openness and system modularity for my coming development projects. In addition, there are still some possibilities for future improvements on top of IPODS. For example, separation of duties to reduce human mistakes and enhance user access securities, data analysis for role optimization, automatic account replication crossover different applications or platforms and XML account portfolios extended from XACML. Therefore, it is concluded that IPODS is worth for implementation and provides the most benefits to CS Dept.
**8.1. References**


### 8.2. Appendix A: Simulation with 1% error rate

**Heuristic Quantitative Model with Stochastic Decision Variables**

**Case 1: Assuming the human error rate is 1%**

<table>
<thead>
<tr>
<th>Decision Variables</th>
<th>Mean (H)</th>
<th>Std dev (σ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Request Frequency (daily)</td>
<td>2</td>
<td>0.5</td>
</tr>
<tr>
<td>Num of System Involved</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>Time (mins) spent for each role assignment</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Time (mins) penalty for each role assignment</td>
<td>30</td>
<td>5</td>
</tr>
<tr>
<td>Percentage of human mistakes or controllable external factors</td>
<td>0.01</td>
<td></td>
</tr>
</tbody>
</table>

#### Calculation Table for the Performance Comparison

<table>
<thead>
<tr>
<th>Replication (3yrs = 1095 days)</th>
<th>Random Number of Request per day</th>
<th>Random Number of System Involved</th>
<th>Random Time used for Role Assignment without any mistake</th>
<th>Random Time used for Role Assignment with human mistakes</th>
<th>Day Request for Role Assignment (95% of 1095 days)</th>
<th>Random Number of System Involved</th>
<th>Estimated Time used in existing situation</th>
<th>Estimated Time used with support of IPODS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1095</td>
<td>2.123975062</td>
<td>5.689044353</td>
<td>2.513243565</td>
<td>4.988456293</td>
<td>9.891394801</td>
<td>5.196089999</td>
<td>10.70729058</td>
<td>11.50972616</td>
</tr>
<tr>
<td>1097</td>
<td>3.210238953</td>
<td>4.239567935</td>
<td>2.319747935</td>
<td>5.189046239</td>
<td>9.213487953</td>
<td>5.156289999</td>
<td>11.31497549</td>
<td>11.45097261</td>
</tr>
<tr>
<td>1098</td>
<td>2.310238953</td>
<td>4.239567935</td>
<td>2.319747935</td>
<td>5.189046239</td>
<td>9.213487953</td>
<td>5.156289999</td>
<td>11.31497549</td>
<td>11.45097261</td>
</tr>
<tr>
<td>1099</td>
<td>4.629746239</td>
<td>2.310238953</td>
<td>2.319747935</td>
<td>5.189046239</td>
<td>9.213487953</td>
<td>5.156289999</td>
<td>11.31497549</td>
<td>11.45097261</td>
</tr>
</tbody>
</table>

**Remarks:** Extra time for resolving role conflicts or the problems caused by some late role assignments.

**Table:**

<table>
<thead>
<tr>
<th>Minimum</th>
<th>Maximum</th>
<th>Mean (H)</th>
<th>Std dev (σ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.723975062</td>
<td>10.8209726</td>
<td>2.123975062</td>
<td>5.196089999</td>
</tr>
<tr>
<td>0.568904435</td>
<td>9.891394801</td>
<td>5.689044353</td>
<td>5.16297549</td>
</tr>
<tr>
<td>0.231023895</td>
<td>10.16297549</td>
<td>3.213471345</td>
<td>10.156289999</td>
</tr>
<tr>
<td>4.988456293</td>
<td>11.50972616</td>
<td>4.239567935</td>
<td>11.31497549</td>
</tr>
<tr>
<td>0.127789723</td>
<td>11.83297054</td>
<td>2.319747935</td>
<td>11.45097261</td>
</tr>
</tbody>
</table>

**Mean (H) and Std dev (σ):**

<table>
<thead>
<tr>
<th>Decision Variables</th>
<th>Mean (H)</th>
<th>Std dev (σ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Request Frequency (daily)</td>
<td>2</td>
<td>0.5</td>
</tr>
<tr>
<td>Num of System Involved</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>Time (mins) spent for each role assignment</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Time (mins) penalty for each role assignment</td>
<td>30</td>
<td>5</td>
</tr>
<tr>
<td>Percentage of human mistakes or controllable external factors</td>
<td>0.01</td>
<td></td>
</tr>
</tbody>
</table>

**Remarks:** Extra time for resolving role conflicts or the problems caused by some late role assignments.
8.3. Appendix B: Simulation with 5% error rate

Case 2: Assuming the human error rate is 5%

<table>
<thead>
<tr>
<th>Decision Variables</th>
<th>Mean (μ)</th>
<th>Std Dev (σ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Request Frequency (daily)</td>
<td>2</td>
<td>0.5</td>
</tr>
<tr>
<td>Num of System Involved</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>Time (mins) spent for each role assignment</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Time (mins) penalty for each role assignment</td>
<td>30</td>
<td>5</td>
</tr>
<tr>
<td>Percentage of human mistakes or controllable external factors</td>
<td>0.05</td>
<td></td>
</tr>
</tbody>
</table>

Calculation Table for the Performance Comparison

<table>
<thead>
<tr>
<th>Replication (3yrs = 1095 days)</th>
<th>Random Frequency Time used for role assignment (Mins)</th>
<th>Estimated Time used in existing situation (Mins)</th>
<th>Estimated Time used with support of IPODS (Mins)</th>
</tr>
</thead>
</table>

Daily Average (Mins): 61.53202892
Standard Deviation: 31.14697183

(Remarks: Extra time for resolving role conflicts or the problems caused by some late role assignments)

Decision Variables

<table>
<thead>
<tr>
<th>Role assignment</th>
<th>Mean (μ)</th>
<th>Std Dev (σ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Request Frequency (daily)</td>
<td>2</td>
<td>0.5</td>
</tr>
<tr>
<td>Num of System Involved</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>Time (mins) spent for each role assignment</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Time (mins) penalty for each role assignment</td>
<td>30</td>
<td>5</td>
</tr>
<tr>
<td>Percentage of human mistakes or controllable external factors</td>
<td>0.05</td>
<td></td>
</tr>
</tbody>
</table>
Part III: Self Evaluation

Different from my other previous projects, our previous supervisors always prefer to know about the final project evaluation rather than my personal evaluation. Along with the project cycle, there are many unusual experiences out of project. These unintended learning outcomes and their implications are valuable to be written as my intellectual assets for future reference. To achieve the goals of self-evaluation, I should evaluate how I judged to the problems and reacted by solution. Therefore, I will mention something about what I have learnt, what kind of difficulties I faced and how I solved them finally in this section.

i. The Core Value of Case Study

At the first meeting of our learning group, I believed that the main reason why we choose IS subject field for our MBA projects was because most of our group members had strong IT working background. Although having some advantages to formulate our sets of recommendations and implementation, I practically didn’t know how to do a case study for MBA project. After finishing this project, I more understood the importance to look more outside the boundary of IT project itself. When I want to be a person working as a senior IT manager would be in the future, I think I should understand more facts surrounding a particular business situation rather than develop solutions to the problem only. To analysis the IT project as business case, it’s not easy
for me to identify the problem at first and evaluate the result after implementation. In the past, the problems meant assignments from senior management and it was no significant value to evaluate the completed tasks in a cost unit. Better than just knowing the terms of case analysis, I actualized what I had studied with this project by defining the problem, formulating the possible alternatives, analyzing the alternatives and recommending a solution with specific action plan. I ensure that this mindset training is very useful for me to understand problems more in-depth and inspire ideas to resolve them much better in future.

ii. More Understanding about Information System

Before starting this project, I have difficulty to distinguish the differences between Information Systems (IS) and Information Technology (IT) even though I have studied MIS course on few months no longer. From my personal interpretation, IS normally is more business-oriented and people-oriented and IT is rather application-oriented. On the latest industrial trends, most of IT workers in HK actually are not working with “IT projects”. However, since most of the project outcomes focused on the service quality of business organizations, the correct names for these projects should be most likely called as “IS projects”. In the service-oriented small city like HK, it rarely provides some project opportunities specified for IT because fewer companies invest on technological researches. Therefore, if the IT workers remain the same mindset as I used previously at the beginning of this project, it is nothing to be surprised why senior management dislike or dissatisfy the results of “IT projects” because their results focus on application
level itself rather than the higher user interaction levels to align with business objectives. Even on my current working situation, eight out of ten should be IS projects. To keep competitive on my quality to manage the coming projects, this project experience gains an insight to know the importance of IS approaches to analyze the real problems at business level.

iii. **Difficulties on Project Evaluation**

Since I had long-term IT project management experience, I was so confident to coordinate everything well, especially on the time management and deliverable quality, from the beginning of MBA project. According to the complexity of project implementation, it was expected that the project couldn’t be finished in the period of MBA project time. I only designed the possible solution and then its implementation would be finished on later. On the other hand, it is difficult to quantify our evaluation result by traditional performance metrics or in dollar values. With the guidance by Dr Fang, I created a new IT effectiveness metrics tailor-made for CS Dept to evaluate the expected outcomes how close to match the key benefits of business objectives, predefined project scopes and user satisfaction. With these benchmarks, my future projects would have a kind of compatible measurement guidelines to know the grey area between success and fail.
iv. Implications on System Openness & Modularity

This is the first time of Web Team in CS Lab to develop a kind of system which provides open platform for others. Previously, many projects were to improve the administrative productivity problems by IT automation. Comparatively, I should spend more time to hear other suggestions and concerns from different stakeholders in CS Dept in order to complete this project. However, it gave me a chance to the survival directions of system development. There are two main points should be remembered in the mind of system and application developers. Under the platform of IPODS, the data sources of access control information come from LDAP directory server. According to the roles of user, IPODS will grant a permission right in XML format known by target application system. From user viewpoints, they just only know to login one application they want, but it actually involves 3 different systems, such as LDAP, IPODS and target server, in behind. It means that some contemporary designs of system applications are modularized into web services like Web 2.0. These modules communicate each others through XML data. Some of system modules, like IPODS, can be data consumer (client) and data provider (server) at the same time. It is difficult to implement this project by a kind of agile development methodologies without detail system specification design at first. I need to spend more time to discuss about the possible communication interfaces between applications. This approach would be valuable to improve the reusability of system application. When developing other system applications on the next time, I would keep in mind that the openness and modularity are relevant design criteria to rise up the synergy effects among existing developed applications.
<END>