出國報告（出國類別：參加國際研討會）

2014 創新科技&工業管理國際研討會

服務機關：國立中正大學資管系
姓名職稱：古政元教授
派赴國家：韓國首爾
出國期間：103 年 05 月 27 日至 103 年 05 月 31 日
報告日期：103 年 06 月 18 日

摘要

此行前往韓國首爾的目的在透過參加世界性之國際學術研討會和國際諸多學者互動討論，並聽取各界學者的建議，冀望能夠增加我們所提出之安全的軟體系統開發生命週期方法的完備性，以供國內產官學研界做參考。未來在相關領域試用後得到的分析結果將用來驗證本系統的可用性，如果成果符合預期的話，我們最後會提交到國際著名的學術期刊上發表。

整個研討會過程本人前往聽取多篇報告，詳細心得將在後述內容呈現，除此之外較大的收穫是我認識了兩位重要的韓國學者，分別是 KEIMYUNG UNIVERSITY 的 Moon-Kyu Lee 教授以及 SEOUL NATIONAL UNIVERSITY OF SCIENCE AND TECHNOLOGY 的 Jung-Sook Lee 教授，我們討論了很多問題，發現彼此間有許多的研究議題相類似，或許以後會有不小的合作空間，也可能邀請兩位中的一位在不久的將來前往台灣訪問，或者是客座一段時間。

另外較重要的成果則是出席本研討會的學者對我們所提出的做法認同，這個發展看來不錯，他們也提供不少實質有用的建議，所以我們會仔細考量如何納入我們的論文當中，最終版本會在挑選的個案經驗證並加在文章內文中後，投稿到適當的國際期刊上，上述學者所提供的很好建議也應該會讓本文的內容更為完整全面。

目次

目的

I. 計畫目標

本計畫欲達成的目標如下：

(1) 應醫療資訊系統之需求，提出一種利用安全型式(Security Pattern)以提升軟體系統開發生命週期安全性的方法，爾後再利用實際的案例以驗證其有效性。

(2) 透過參加相關的國際研討會，與國際學者互動討論，期能夠增加本成果的完備性，藉以提供國內產官學研界做參考。

II. 主題

當政府和企業專注於電子病歷與醫療資訊系統的發展時，潛在的資訊安全問題浮出水面，由於受到個資法的要求，如果不能確保資料安全的話，醫療院所將始終只能戰戰兢兢的使用資訊科技，不知何時會被病患告上法庭。事實上醫療資訊系統及電子病歷的安全問題又較一般企業資訊系統複雜許多，這是因為醫療行為的特殊性及即時性，所以系統的開發者常會聚焦在流程配合上而忽略安全性的問題。然而如果是系統開發完成後才來處理安全問題的話將會事倍功半，吃力不討好。所以本研究的主題將集中在解決：軟體系統開發生命週期的安全問題，而我們所提出的方法是妥善利用安全型式(Security Pattern)的導入。

III. 緣起

軟體安全是產業和研究學界的重大課題，在醫療行業更是如此，這是因為醫療產業所處理的資料是個人最私密的訊息，而絕大多數人最不願意洩漏的也就是個人健康醫療的資訊。電子病歷系統（EMR）的目的是建立一個堅實的基礎設施，以實現健康信息交換（HIE），然而必然伴隨而來的是很多安全問題的潛在威脅將會產生，這亦即所謂的軟體內在安全漏洞。為了有效地克服這一障礙，合理的方法是應用軟體結構中特定的模式與方法。因此，我們經過許久的收集文獻與研究之後，提出一種利用安全型式(Security Pattern)以提升軟體系統開發生命週期安全性的方法。這些安全型式(Security

Pattern)可以使得軟體開發人員能夠將安全問題納入了軟體開發生命週期（SDLC）當中來考量並處置。資訊安全的目的是用來實現資訊的安全屬性（例如：保密性，完整性和可用性...等），目前我們的社會高度依賴分散式運算與處理，網路分散式系統也就意味著確切地保證這種系統的安全性是至關重要的。一般來說，軟體安全的重任往往落在設計和開發者手上，惡意程式感染仍然是最常見的攻擊，駭客也常常以個資為攻擊的標的，但不幸的是，許多醫療組織並未重視或建立與實施安全的軟體開發生命週期（SDLC）以保護他們的資訊資產。因此，軟體的安全性應當要被特別的強調，合適的對策也該被提出來才是。根據 HIMSS AnalyticsTM 電子病歷採用模型顯示，電子病歷的最終目標是實現健康訊息交換（HIE）。然而，它也激起患者的安全和隱私問題，醫療服務提供者在使用上也常有顧慮，即使病歷交換可以讓病患及醫院皆受益，可是如何使用並在醫院共享病歷以提高醫療品質，同時又不違反病人的隱私則是一個很重要的考量。此外，隱私和 HIPAA 安全規則要求確保執行安全政策及控制措施和技術，以保護醫療院所的資訊基礎設施，並監測和控制區域內和跨組織間訊息的傳送。因此，提高醫療系統軟體的安全性的品質是一個非常明顯且必然的趨勢。

對於開發者和使用者而言，修復系統漏洞以提高系統的安全是需要很大的成本並且相當昂貴。為了避免這個問題，提前考慮安全問題是非常必要的。在另一方面，許多開發商試圖通過參考最佳實務或基準的解決方案來提高軟體與系統的安全性，然而這些做法並不經常一體適用，仍然常常需要有客製化的做法。此外，安全的問題，不應僅僅聚焦在實施和部署階段，應考慮軟體開發生命週期中的每個階段。目前，很多最佳的安全實務較專注於執行和部署問題，所以無法解決前面所介紹的各種安全漏洞。安全型式(Security Pattern)提供了一種解決方案，考量特定範圍內所產生的安全問題，支持開發人員避免設計出有系統漏洞的軟體程式，並在軟體開發生命週期（SDLC）過程中提高軟體的結構性。此外，安全型式(Security Pattern)已被許多的最佳實務和典範標準認定其可用性。由於上述這些優點，本研究決定採用安全型式(Security Pattern)的概念，旨在提高軟體架構的安全，加上安全型式(Security Pattern)是可重複使用，軟件開發者也可以減少開發系統的時間與預算以及伴隨著安全漏洞的風險成本。

IV. 效益、達成事項及未來規劃

本計畫在參加 2014 創新科技&工業管理國際研討會時，和許多來自各國的學者一起討論，並且獲得一些寶貴的建議，藉以修訂如何最佳地使用安全型式(Security Pattern)以加強在醫療環境中的系統開發生命週期的安全，期能更加確保醫療體系中病人的資料無虞，爾後我們將再找尋適當的醫療專案以驗證所提出方法的有效性，及至分析結果數據完備之後，未來我們還計劃將最終版本提交到國際著名的學術期刊上發表。

過程

到達 2014 年科技創新與產業管理國際會議 Conference on Technology Innovation and Industrial Management 的會場後，我立刻積極開始和一些學者與同儕交流做研究的經驗和意見，如附錄一圖一及圖二的照片所示，也聽取不少的論文報告。本次會議的官方會議議程如下：

Schedule of 2014 International Conference on Technology Innovation and Industrial Management
28th-30th May 2014, Seoul, South Korea

Tuesday, 27 May 2014		
Time	Description	Location
18:00 – 19:30	Registration & Welcome Reception Professor Tack Hyun Shin Dean, College of Business and Technology Seoul National University of Science and Technology	Holiday Inn Seongbuk
Wednesday, 28 May 2014		
8:00 – 9:30	Registration	SEOUL TECHNOPARK
9:30 – 10:15	Conference opening by Hosting University • Video Show for SeouTech • Welcome Address by Dr. Keun Namkoong President, Seoul National University of Science and Technology Welcoming address by TIIM Honorable Executive Committee (based on the previous hosting university) • Dr. Binshan Lin Louisiana State University in Shreveport, USA • Dr. Pekka Kess Oulu University, Finland • Professor Ryszard Debicki, Maria Currie-Sklodowaska, Poland • Dr. Bordin Rassameethes Kasetsart University, Thailand	SEOUL TECHNOPARK Smart Hall
10:15 – 10.45	Keynote speaker: Dr. Yongtae Park, Professor of Industrial Engineering, Seoul National University, Korea Topic: "On the Emerging Paradigms in Technological Innovation: Sustainable or Perishable?" **[PDF]**	SEOUL TECHNOPARK Smart Hall
10:45 – 11:10	Morning break	
11:10 – 11:40	Keynote speaker: Dr. Jong Guk Song President, Science and Technology Policy Institute, Korea Topic: "Innovation Policy of Korea in the Creative Economy" **[PDF]**	SEOUL TECHNOPARK Smart Hall
11:40 – 12:00	TIIM 2015 Presentation, Slovenia	SEOUL TECHNOPARK Smart Hall
12:00 – 13:00	Lunch	
13:30 – 15:30	6 Parallel Sessions: 20 minutes per paper	SEOUL TECHNOPARK
15:30 – 15:50	Afternoon Break	
15:50 – 17:50	6 Parallel Sessions: 20 minutes per paper	SEOUL TECHNOPARK
18:30	Reception and Special Event (Taekwondo Performance and Korean Drum Performance)	100th Memorial Bldg Moksan Gallery Hall

Schedule of 2014 International Conference on Technology Innovation and Industrial Management
28th-30th May 2014, Seoul, South Korea

Thursday, 29 May 2014		
9:30 – 10:40	**Rector Forum** "Managing University in Competitive Environment through Sufficiency Growth"	SEOUL TECHNOPARK Smart Hall
10:40 – 11:00	Morning break	
11:00 – 12:00	**Editors' Panel** "Publication in International Journal, an insight from journal editors"	SEOUL TECHNOPARK Smart Hall
12:00 – 13:00	Lunch	
13:00 – 15:00	6 Parallel Sessions: 20 minutes per paper	SEOUL TECHNOPARK
15:00 – 15:20	Afternoon Break	
15:20 – 17:00	6 Parallel Sessions: 20 minutes per paper	SEOUL TECHNOPARK
19:30 – 21:30	Farewell Dinner at Han River Cruise	Move by Bus
Friday, 30 May 2014		
8:00 – 14:00	Company Visit – Kia Motors Plant at Gwangmyeong City, Bus Pick-up Point: Holiday Inn Seongbuk	

I. 議場主題

本次國際學術研討會的主題仍舊沿用過往的多種學科領域，包括：會計資訊科技、品牌價值和管理、變更管理、企業融資、企業，運營和生產策略、文化多樣性、顧客心理、客戶關係管理、電子商務與電子商業、電子學習和人力資源、電子和行動政府、企業和營運風險、綠色技術和生產力、資訊管理與電腦安全、創新管理、國際業務及市場、投資、知識管理、管理及企業發展、管理資訊系統、營銷策略和管理、併購和收購、動機與情緒智惠、網路政府、新產品和服務開發、一站式服務、組織心理學、績效衡量與管理、電子商務中之隱私和安全問題、生產技術、管理和改善生產力、宣傳媒體、公共價值、品質改進和管理、社會營銷、供應商夥伴關係和供應鏈管理、持續經濟、經營和產業化經營技術、加值管理等。但今年更聚焦在技術創新與工業管理的融合、和諧、多樣性和可持續性等議題上，這其實跟近年來產學研所最有興趣的綠色議題有非常密切的關係。事實上在去年我就已經注意到這個趨勢，我記得去年有一個報告者談到綠色供應鏈，他認為許多人仍然誤解綠色供應鏈不必然會改善效率和降低成本，但事實上現今綠色供應鏈強調兩個主要標竿並不與企業目標違背，他也觀察到: (1)越來越多企業將綠色供應鏈管理的目標與業務目標看齊; (2)使用綠色供應鏈是改善企業流程的企機。希望我們國家能夠注意到這個很明顯的全球趨勢，並且投入更多資源在這個產學領域上。

我的演講定在 5 月 29 日下午 13:00 的場次，這個場次的主題是：決策理論、數學模型與統計 – 程式設計、軟體發展與應用。

II. 個人所發表內容摘要、現場報告或討論交流情形

我提早許多時間抵達報告場地，雖然不是第一個報告，但我早早就將 power point 上載妥當，本場次開始時有四個報告人出席，我是第二位報告者，現場如附錄一圖三、圖四及圖五的照片所示。

我的報告題目是一種利用安全型式(Security Pattern)以提升軟體系統開發生命週期安全性的方法，尤其是要針對醫療產業的需求做考量。安全型式(Security Pattern)被廣泛地用來解決變化萬千的安全問題，從架構層次的模式到系統的設計，都分別提供了許多如何實用在模組中的方法。安全型式(Security Pattern)是專有名詞，它包含下述元件：名稱、其他名稱、例子、背景、問題、解決方案、結構、動態、履行、案例、變種、已知應用、後果、其他等。它們也有許多不同的分類，包括用在結構、行為、網絡、主機、應用程序或者功能類型。但是，使用這些安全型式(Security Pattern)的標準過程似乎仍然不全然可行，所以基於軟體系統安全的目的，我們已經發展出一套應用安全型式(Security Pattern)以增加安全度的系統開發生命週期步驟，接下來的重要工作將是如何在醫療產業中驗證其效用。

聽眾們多留下非常深刻的印象，除了讚許之外，他們還提供了有用的建議，讓我未來有改善的方向。會後我有跟一位國外學者討論，其中一個最重要的建議是，提醒我應當繼續注意效用的驗證，因為這些驗證的結果將強烈影響讀者對我們所提出方法的信心，他建議我們不只要找一個醫療組織或單位來實證，最好要有更多的使用單位或組織提出使用後的感受或感覺才會更有說服力。他也特別強調這個議題或將不只是技術問題，管理與追蹤機制也是相當必要的，否則的話，眾多開發者是否都有全然遵循安全型式(Security Pattern)所建議的作法與過程並不太容易確認。事實上，再好的標準與架構如果只是當個參考而沒有完全遵守的話，這跟完全不用有時沒有太大的差別，因為安全的議題有時候得到 90 分跟 0 分是差不多的，因為同樣都有漏洞可能會被利用。

III. 聽取報告議題之內容重點摘述、見聞或新知

在我參與的場次中，主席 KEIMYUNG UNIVERSITY 的 Moon-Kyu Lee 教授談到這些年來，工業上的大量生產常需要在一個大的薄板上進行零配件的切割，例如:皮革，橡膠，紡織，塑料，木材，金屬板材等。因為這種切割運動的複雜性，大多數這樣的切削操作都不可避免地得在電腦數值控制（CNC）機械加工系統的幫助下才能完成。為了降低使用 CNC 機床的時間，提升生產的效率，如何減少刀具路徑就成為一個相當重要的問題。

他所研究的切割路徑最佳化問題是要在一個板塊上，在每個開放和封閉輪廓的材料形狀裏決定最優的切割序列。因此，需要對庫存板的輪廓和其入口點的切割序列同時進行優化，以使得切割的總非生產性行進距離最小。文獻中許多學者已提出各式各樣解決問題的方法，但大多數先前的研究中考慮的為庫存板，其中部分切割被簡單地由一個單一的封閉或開放的輪廓來表示，然現實的問題是，在孔和開放和封閉類型的內部輪廓間的移動有無限多種組合，這是不容易完整全面考量的。因此 KEIMYUNG UNIVERSITY 的 Moon-Kyu Lee 教授提出一種新的方法，該方法是基於混合遺傳演算法（HGA）結合 r 選項啟發式輪廓序列的局部最優和動態規劃演算法（DP）來確定最佳穿刺點。從計算結果可以觀察到如下現象：

1.該 HGA 產生與搜索的 r 選項有足夠深度使得每次運行可以得到一個確切的最佳解

2.區域搜索的完整深度是不需要 r 選項啟發式，因此 HGA 效率的提高是可以實現的

由於本文處理的問題是非常全面的，該演算法可廣泛應用於各種數值控制基礎的切割工藝上，例如雷射、火焰、電子束和水刀加工等。

此行的最大收穫是認識了這位來自韓國 KEIMYUNG UNIVERSITY 的 Moon-Kyu Lee 教授，他目前在工業與系統工程系任教，他的專長非常廣泛，主要專注在工業生產最佳化的所有相關研究議題上，我們討論了一些問題，我打算邀請他在不久的將來前往台灣訪問我的學校，或者是客座一個學期。此外這次參加國際研討會的另一個收穫是也認識了韓國 SEOUL NATIONAL UNIVERSITY OF SCIENCE AND TECHNOLOGY 的 Jung-Sook Lee 教授，她的研究領域也一樣相當廣泛，但是更偏向實務面上的應用，主要是資訊系統導入之所有相關的各項諮詢與協助，或許這也是我未來可以合作的對象之一。

心得及建議

參與此次 2014 年科技創新與產業管理國際學術會議 Conference on Technology Innovation and Industrial Management，著實有不少的收穫，也讓我留下深深的印象。通過與來自世界各地的專家和學者討論交換觀點，我真的是得益於他們的嶄新看法。台灣的醫療產業逐步走向國際化，電子病歷和資訊系統也更進一步加速這個產業許多層面的進步，但也讓我們面臨許多資訊安全上的威脅。事實上，台灣的醫療產業與技術具有亞洲或全球的領先地位，在世界上也成為許多國家的學習對象。但是，如果我們要更上一層樓，我們必須要特別注意資訊系統與高精密醫療設備正在如何改變醫療的行為。高速網路化與交換技術以及資訊系統使得跨國醫療變得很容易，但也更需要能夠保證各種安全上的要求，否則大多數的醫院/組織將不敢採用它。我們的研究針對醫療業需求提出了一種利用安全型式(Security Pattern)以提升軟體系統開發生命週期安全性的方法。這份研究報告與醫療產業的資訊安全問題有關，實際上，它被認為是影響電子病歷發展的最重要因素之一。因此，這次演講吸引了很多目光焦點，我們確實也有一些討論，學者對我們所提出的運作架構認為殆無疑義，只是對後續的發展有如前所述的建議。這個初步發展看起來是不錯的，我們論文的最終完整版本將結合其間一些學者所提供的建議，以便使其更為完整全面。我還計劃將最終版本提交到國際著名雜誌發表，相信這也是參加這個國際學術會議最顯著的收益。

在參加此次國際會議的討論中，我有以下的重要心得。因為會議的關係我們得以觀摩到韓國大學的創新育成中心，創新育成中心係降低中小企業創業之阻力與科技研發資源不足之窘境，希望藉由學校師生與設備來開發較低成本之技術方法，再利用中心這個平台擴散至業界，最後形成國家的競爭力。所以一個國家在大專院校所建立或協助建立的創新育成中心就具有非常重要的任務，成功與否也會極大的影響國家產業未來的發展。這次我們看到韓國大學的創新育成中心，其規模之大、經費之多及人力之充沛等皆非國內大學所能望其項背，然後細數他們中心所產出的成果也覺得真的很令人敬佩。雖然韓國重要的經濟活動是由大企業所主導，多數 GDP 也是由大企業所貢獻，但是不代表他們不扶植創新中小企業。所以我建議政府及教育部可以參考韓國的創新育成中心的政策，再加上我國本來在中小

企業的發展上就是強項，因此經驗非常豐富，只要有充分的資源投入，適當政策的引導，而目前國內的高等研發人力也非常足夠，相信再有政府的介入協助之下，或能創出一段創新企業的另一歷史高峰也說不定。

在這次的報告中，我們提出了一種利用安全型式(Security Pattern)以提升軟體系統開發生命週期安全性的方法。安全型式(Security Pattern)被廣泛地用來解決變化萬千的安全問題，從架構層次的模式到系統的設計，都分別提供了許多如何實用在模組中的方法。安全型式(Security Pattern)是專有名詞，它包含下述元件：名稱、其他名稱、例子、背景、問題、解決方案、結構、動態、履行、案例、變種、已知應用、後果、其他等。它們也有許多不同的分類，包括用在結構、行為、網絡、主機、應用程序或者功能類型。但是，使用這些安全型式(Security Pattern)的標準過程似乎仍然不全然可行，所以基於軟體系統安全的目的，我們已經發展出一套應用安全型式(Security Pattern)以增加安全度的系統開發生命週期步驟，這是針對醫療產業需求所設計的架構，接下來的重要工作將是如何在醫療院所當中驗證其效用。本研究嘗試為醫療產業的軟體系統開發生命週期安全投入心力，我們也希望這結果能夠引導未來在這一領域的研究。

附錄一

圖一 Keynote Speech 結束後離開會場前

圖二 與韓國 SEOUL NATIONAL UNIVERSITY OF SCIENCE AND TECHNOLOGY

Prof. Jung-Sook Lee 合影

圖三 Presentation 1

圖四 Presentation 2

圖五 與韓國 KEIMYUNG UNIVERSITY Prof. Moon-Kyu Lee 合影

附錄二

A Medical Software Development Framework Based on Security Patterns

Man-Nung Liu, Cheng-Yuan Ku, and Tsung-Han Yang

Department of Information Management,
National Chung Cheng University,
Chia-Yi County, Taiwan, R.O.C.

Introduction (1/3)

- The high dependence of our society on distributed and networked systems means that the assurance of security for the above-mentioned systems is important
- In the early days, software was often designed and developed without the concerns of security in the mind of developers
- Even until now, few organizations establish and implement the formal secure system development life cycle (SDLC) to protect their information systems

Introduction (2/3)

- Required by laws or regulations in many countries, the medical systems especially need the security measures since they are highly involved in the privacy and security of patients
- Electronic medical record (EMR) => Health information exchange (HIE)
  - the dilemma between sharing medical records across hospitals and securing patients' privacy emerges

Introduction (3/3)

- Therefore, it is really necessary to reduce the risk accompanied with vulnerabilities in the medical systems
- Security concerns should be addressed in all phases of SDLC, not just in the implementation and deployment phases
- A framework of adopting security patterns for the medical system is proposed to avoid system vulnerabilities, address security issues, and improve the software architecture

THEORETICAL BACKGROUND

Security Concerns in SDLC

- From literature, in spite of the well-planned SDLC process, the security problems still exist in some of the developed systems
- Security concerns must be carefully addressed
  - in the requirement phase
  - in the design phase
  - in the implementation phase

In the Requirement Phase

- Control gates
  - determination of acquisition strategy
  - enterprise architecture (EA) alignment
  - system concept review
  - performance specification review
  - risk management review
  - financial aspect review

In the Design Phase

- Control gates
  - architecture/design review
  - system performance review
  - system functional review
  - project status and financial review
  - follow-on review of risk management decisions

In the Implementation Phase

- Control gates
  - plan and conduct system certification and accreditation
  - test security controls
  - review final project status

Security Pattern

- Security patterns address security issues in various industries
- No formal and systematical use is suggested for SDLC

Patterns

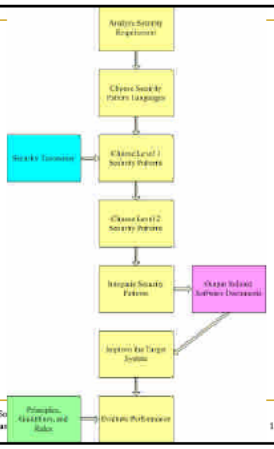
- Advantages of pattern
  - helping build complex and heterogeneous software architectures
  - supporting the construction of software with defined properties
  - providing a means of documenting software architectures
  - documenting existing and well-proven design experience
  - providing common design principles
  - identifying and specifying abstractions of software components
  - addressing recurring design problems and presenting solutions to them

Patterns in the Medical Industry

- Patient Treatment Pattern
- SOAP Pattern for Medical Charts
- Sistemas de Gerenciamento de Clinicas in portuguese (SiGCli)
- No security pattern is considered for use in the medical industry

Proposed Framework

The Process of Adopting Security Patterns



Analyze Security Requirement

- Identification
- Analysis
- Mapping
- Documentation
- Review and Verification

Choose Security Pattern Language

- Consideration
  - security requirement specification
  - scope, classification and integrity of pattern language
- Decide the design techniques based on the chosen pattern languages

Choose Security Pattern in Level 1

- Security taxonomy
  - choose matched security areas
  - construct the relationships of security elements
  - organize secure software architecture
- Based on security taxonomy, the user decides which categories of security concerns should be adopted

Choose Security Pattern in Level 2

- Detailed problem-solution pairs
- Consider the chosen pattern one by one

Integrate Security Patterns and Improve the Target System

- By using UML diagrams
- Use Case Diagram to illustrate the major functions
- Activity Diagram and Sequence Diagram to describe the interactions between components
- Class Diagrams to compare the status
  - before security control
  - after security control

Evaluate Performance

- Principle
  - the general security wisdom on the philosophy level
- Guideline
  - the recommendation about something to do or avoid on the semantic level based on security
- Rule
  - the recommendation about something to do or avoid on the syntax level

Future Work

- Use one or more cases to self-validate our framework
- Security analysis
- Performance analysis

Thanks for Your Attention

附錄三

Thursday, 29 May 2014

Sessions 5C: Lecture Room 304

附錄四

Proceedings of 2014 International Conference
on Technology Innovation and Industrial Management
28th-30th May 2014, Seoul, South Korea

附錄五

Proceedings of 2014 International Conference on
Technology Innovation and Industrial Management
28th-30th May 2014, Seoul, South Korea

A SECURE SOFTWARE DEVELOPMENT FRAMEWORK BASED ON SECURITY PATTERNS

**Man-Nung Liu, Department of Information Management,
National Chung Cheng University, Taiwan, R.O.C.
lmn198994@yahoo.com.tw**

**Cheng-Yuan Ku, Department of Information Management, National Chung Cheng University, Chia Yi County, Taiwan, R.O.C.
cooper.c.y.ku@gmail.com**

**Tsung-Han Yang, Department of Information Management, National Chung Cheng University, Chia Yi County, Taiwan, R.O.C
h730615@hotmail.com**

ABSTRACT

Information security, a field of concepts and techniques for achieving security properties (e.g. confidentiality, integrity, and availability...etc.), is of great importance in the technology-advanced world. Furthermore, our critical dependence on highly distributed networking systems means that accurately assuring the security of such systems is really essential since the isolation is not possible (Lipson and Weinstock, 2008). Generally, the software is often designed and developed without security being in the mind of the developers (Viega and McGraw, 2001). As well known, malware infection continues to be the most commonly seen attack but few organizations establish and implement the secure System Development Life Cycle (SDLC) to protect their information assets (Martin and Rice, 2011). Therefore, software security should be further emphasized, and a formal development framework is necessary.

The production of a software system is usually conducted through the SDLC (Moore, 2008). However, in spite of adoption of well-planned SDLC process, many developed systems are still not free from security concerns. Hence, security concerns must be considered during every phase of software development, from requirement fetching, design, implementation, testing until deployment (Devanbu and Stubblebine, 2000).

From the perspective of software development, pattern or design pattern is a written document that describes a general solution for a frequently occurred problem. A problem and

its general solution are combined as a problem-solution pair and its common factors lead to patterns. Software designers can refer to these design patterns to find proper solutions for their systems.

Security patterns address security issues at widely varying levels of specificities ranging from architectural-level patterns involving the high-level design of system to the implementation-level patterns providing guidance on how to implement portions of functions or methods in the system. Many different classifications of them are developed in the existing studies, such as aspect types (creational, structural, or behavioral), abstraction level (network, host, or application) (Konrad et al., 2003), and function types. However, a standard procedure of using these security patterns seems still unavailable.

*For easy understanding and convenient communication, patterns are designed with fixed formats that provide systematical information to adopters. According to Pattern-Oriented Software Architecture (*Buschmann *et al., 1996), the general pattern format is described in the following Table 1. Based on the security patterns, we have figured out the initial procedure of secure software development.*

Table 1 Pattern Structure

Component	Description
Name	The name and a summary of the pattern.
Also Known As	The other names of the pattern.
Example	A real case demonstrating the problem and the need of the pattern.
Context	The situations in which the pattern may apply.
Problem	The problem the pattern addresses, including a discussion of its associated forces.
Solution	The fundamental solution principle underlying the pattern.
Structure	A detailed specification of the structural aspects of the pattern, using appropriate notations.
Dynamics	Typical scenarios describing the run-time behavior of the pattern.
Implementation	Guidelines for implementing the pattern.
Example Resolved	Discussion of any important aspects for resolving

Component	Description
	the example which are not yet covered in the Solution, Structure, Dynamics, and Implementation sections.
Variants	A brief description of variants or specializations of a pattern.
Known Uses	Examples of the use of the pattern, taken from existing systems.
Consequences	The benefits the pattern provides, and any potential liabilities.
See Also	References to patterns that solve similar problems.

Until now, we still continue improving this procedure. Moreover, the practical test is under preparation. That is, we will adopt this method to assist in the development of a software system at a case institute and then analyze the efficiency of this proposal. Moreover, we will also derive the managerial insight to enhance our contribution.

CONCLUSIONS

Software security is always a major issue for both the industrial and research communities during these years. Accompanying with the inevitable trend, a lot of potential threats may result in the financial or intangible loss of many software adopters. To effectively overcome this obstacle, the reasonable way is to apply specific patterns to software development architecture. Hence, this study is mainly aimed at proposing a process with security patterns to implement the well-structured software. These patterns enable the managers of software projects to incorporate the security concerns into the SDLC in order to raise the security level.

Keywords: Security Patterns, Software Security, Software Development Life Cycle

REFERENCES

1. Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., and Stal, M. (1996). Pattern Oriented Software Architecture Volume 1: A System of Patterns. NJ: John Wiley & Sons.
2. Devanbu, P. T. and Stubblebine, S. (2000, May). Software engineering for security: a roadmap. In: Finkelstein, A. (Ed.), Proceedings of the Conference on the Future of Software Engineering, NY: ACM, 227-239.

3. Konrad, S., Cheng, B. H., Campbell, L. A., and Wassermann, R. (2003, March). Using security patterns to model and analyze security requirements. In: Bay, M. (Ed.), Proceedings of the 2nd International Workshop on Requirements Engineering for High Assurance Systems (RHAS '03), California: IEEE.
4. Lipson, H. and Weinstock, C. (2008). Evidence of assurance: Laying the foundation for a credible security case. Pennsylvania: Carnegie Mellon University.
5. Martin, N. and Rice, J. (2011). Cybercrime: Understanding and addressing the concerns of stakeholders. Computers & Security, 30(8), 803-814.
6. Moore, J. W. (2008) ISO 12207:2008. Systems and software engineering - Software life cycle processes. Geneva: International Organization for Standardization
7. Viega, J. and McGraw, G. (2001). Building Secure Software: How to Avoid Security Problems the Right Way. Boston: Addison-Wesley.