특집

CBD환경을 위한 4GL 어플리케이션의 웹 어플리케이션으로의 변환

박 병 형[1] 양 해 술[2]

목 차

1. 서 론

1.1 타 산업에서의 재활용

산업이 고도로 발달하면서 자원에 대한 재활용성이 지대한 관심을 끌고있다. 1차 산업의 경우에도 완제품의 생명이 끝나는 시점에 기계를 분해하여 재활용할 부품들을 찾아내고 재가공한다. 이렇게 재활용이 가능한 것은 제품을 부품중심으로 만들었기 때문에 가능한 일이다. 깨어진 도자기를 재사용하는 것은 불가능하지만 여러 가지 기능들을 각각의 객체로 분리시켜서 조립된 기계제품은 부품 하나하나의 기능이 필요한 곳에 재배열되고 재활용될 수 있다. 이러한 자원의 재활용은 주변 환경의 변화에 보다 용이하게 적응할 수 있고 2중 생산의 낭비를 예방하여 비용절감효과를 가져온다. 이러한 메커니즘은 하드웨어에만 적용되어 왔지만 최근 소프트웨어에도 재활용의 메커니즘이 적용되고 있다.

1.2 S/W의 재활용 추세

이러한 소프트웨어의 재활용 연구는 객체지향적 사고를 기반으로 하고 있으며 물리적인 것 외에 논리적인 소프트웨어에 객체지향적 사고를 적용시킨다는 것은 힘든 일이 아닐 수 없다.

하지만 많은 기업들은 소프트웨어를 자산으로 여기기 시작하였고 소중한 자산을 최대한 활용하기 위해 소프트웨어에도 재활용 및 재사용 개념을 적용할 수 밖에 없다는 것을 깨닫고 있는 시점이다. 따라서 현재까지 가장 널리 쓰이고 있는 4세대 프로그램 언어(이후 4GL)를 객체지향적 시각에서 어떻게 최신의 컴포넌트 기반 시스템으로 변환해야 할 것인지에 대한 연구를 하게 되었다.

4GL 어플리케이션을 CBD환경의 Java어플리케이션으로 변환함으로써 기업의 소중한 자산인 S/W자원의 체계적이고 합리적인 관리를 가능하게 하고 세계적인 IT 표준화에 발 맞추어 적시에 고객의 요구에 답할 수 있게 하여 결과적으로 기업의 경쟁력 향상을 위한 연구다.

2 관련연구

2.1 4GL 재활용성

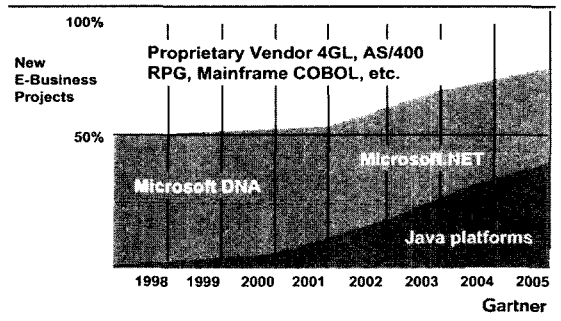
1) 호서대학교 벤쳐전문대학원 컴퓨터응용기술학과 박사과정
2) 호서대학교 벤쳐전문대학원 교수

2.1.1 4GL 어플리케이션의 특징

다양한 어플리케이션 중에서 왜 4GL 어플리케이션을 주제로 다루었는가는 앞서 서론에서 언급하였다. 컴퓨터 하드웨어의 급속한 발전으로 응용업무의 생산성에는 수십 배의 생산성 증가가 필요하였고 이에 따라 4세대 이전의 언어들에 비해 비전문가들도 쉽게 접근할 수 있고 강력한 생산성을 지닌 프로그래밍 언어가 필요하게 되었다. 이는 사용자에게 즉, 사람에게 더욱더 친숙한 언어를 말하는 것이다. 4세대 언어는 프로그래머가 입력한 값을 의사코드로 변환시키고 이것을 다시 기계코드로 고쳐 실행시키는 특징을 가지고 있다. 그리고 사용자에게 친숙한 환경을 제공하기 위해서 최대한 자연어에 가까운 문법을 사용하도록 하고 불편한 문법을 배제하였다. 기존 어플리케이션에서 CBD환경의 어플리케이션으로 변환하는데 있어 4세대 언어가 가지는 중요한 특징은 화면과 데이터베이스 등의 구분이 이전의 언어보다 상당히 명확하다는 것이다.

2.1.2 변환의 이유(Trend)

최근 인터넷의 등장하였을 때 만큼이나 IT 산업에 커다란 변화가 오고있다. 바로 웹 서비스(Web Service)라는 변화다. 인터넷의 등장에 따라서 기존의 Client/Server 시스템에서 웹 이너블(Web-enable)기술을 적용하게 되었다. World Wide Web을 통하여 세계 어디서나 업무를 볼 수 있게 되었으며, 기존에는 기업 내에서만 적용해 왔던 시스템을 B2C를 통해 고객과의 커뮤니케이션이 가능하게 되었다. 그리고 B2B. B2G를 통하여 기업과 기업, 기업과 정부의 네트웍이 이루어 질 수 있게 되었다. 하지만 과도한 부하에 따른 2-tier의 웹 이너블(Web-enable)시스템은 한계를 보이기 시작하였다. 그러한 문제점에 대한 대안은 분산시스템의 도입으로 과도한 부하를 처리할 수 있다. 웹 서비스 표준을 따르는 J2EE플랫폼은 웹 어플리케이션 서버(Web Application Server)를 사용하여 분산처리를 지원하고 EJB(Enterprise Java Beans)라는 소프트웨어 컴포넌트를 이용한 CBD환경의 원활한 유지보수 및 Transaction처리를 지원한다. 또한 EJB와 XML을 통한 웹 서비스가 가능하다. 웹 서비스를 통해서 인터넷을 통해서 기업과 기업, 기업과 정부, 기업과 고객들의 광범위하고 고품질의 서비스를 가능하게 되었다. 위와 같은 이유로 최근의 어플리케이션 개발은 대부분 CBD환경의 플랫폼인 J2EE 또는 .NET플랫폼으로 개발되고 있는 실정이다.



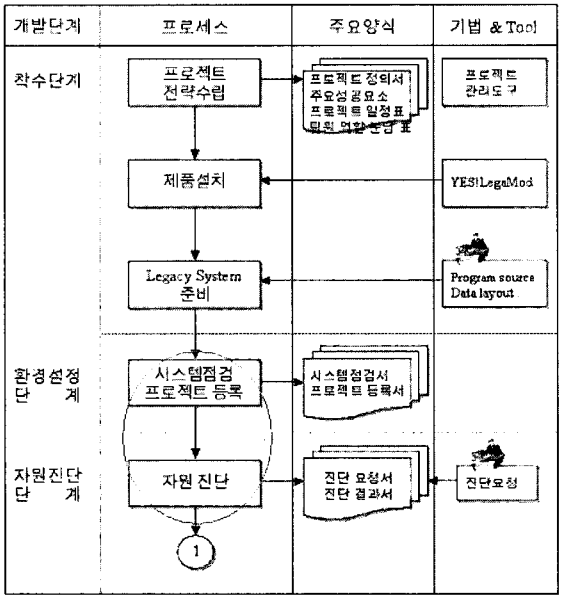
(그림 1) 새로운 e-Business 개발추세

지금까지의 관례대로라면 기존에 기업이 보유하고있는 시스템은 별개로 취급하고 다시 CBD기반의 어플리케이션을 구축하는 방식을 취할 것이다. 하지만 일반적으로 기업의 어플리케이션에는 일반업무에 관한 프로세스와 그 기업의 고유한 프로세스가 녹아있다. 따라서 기존의 구축되었던 어플리케이션도 기업의 고유한 업무프로세스를 포함하고 있고 그러한 업무프로세스는 CBD환경에서도 거의 동일하게 적용된다고 볼 수 있다. 그러므로 프로그램의 언어적인 특성보다는 그 어플리케이션 속에 포함되어있는 비즈니스 프로세스를 찾아내어 새로운 아키텍쳐에 적용시키는 일이 가장 효율적인 방법이라고 할 수 있다.

2.2 CBD기반의 Java어플리케이션으로의 변환

2.2.1 변환의 사전작업

일반적으로 어플리케이션을 구축하기 이전에 수행되어야 할 작업에는 분석/설계가 선행되어야 하듯이 변환을 하기 위해서도 사전작업이 필요하다. 기존 자원을 파악하고 목표로 하는 아키텍쳐에 맞는 자원을 재구성하기 위한 사전작업이 필요하다.



(그림 2) 사전작업의 절차

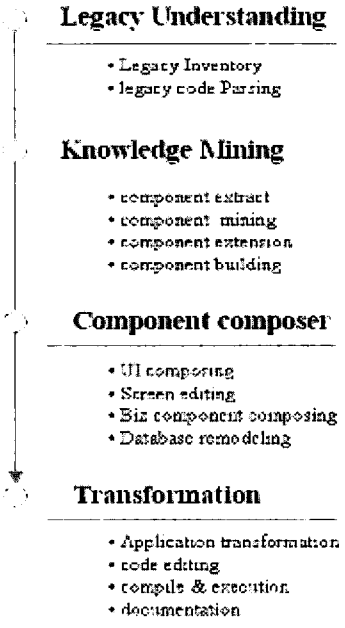
2.2.2 변환의 절차

먼저 기존 어플리케이션의 소스코드를 분석하여 현황을 이해하는 'Legacy Understanding' 단계를 거친다. 기존 시스템의 소스코드를 분석해서 어플리케이션의 자원현황을 List out하고 CBD 환경의 아키텍쳐에 맞는 컴포넌트로 재구조화를 준비하는 사전단계에 해당된다.

두 번째로 분석된 기존 어플리케이션에서 업무 프로세스 등 지식을 추출하는 단계가 있다. 화면, Business Logic, Database를 추출하고 이전 어플리케이션의 형태와 비교하면서 새로운 타겟 어플리케이션의 아키텍쳐에 적용하여 표준 EJB컴포넌트화 하고 이를 정재하고 확장한다. 이것을 'Knowledge mining' 단계라고 한다.

세 번째로 이렇게 생성된 컴포넌트들을 조립한다. 생성된 UI, Business process, Database 컴포넌트들의 조합으로 어플리케이션을 새로운 아키텍쳐로 이전한다.

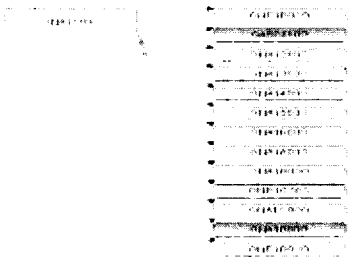
마지막으로 4GL로 짜여진 소스코드를 구문분석을 통하여 Java 소스코드로 Line by line 변환하는 'Transformation' 단계를 거친다. 소스코드의 변환은 마지막 단계에서 순차적으로 진행되는 것이 아니라 컴포넌트의 추출, 조립하는 과정에서 선행된다.



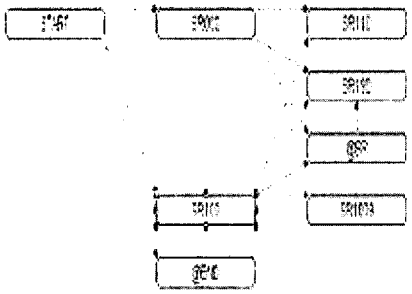
(그림 3) 변환절차도

2.2.2.1 Legacy understanding

아래 그림은 기존 시스템의 자원현황을 분석한 결과이다. 왼쪽의 (그림 4)는 프로그램에서의 호출관계를 나타내는 'Call graph' 이고 오른쪽의 (그림 5)는 프로그램내의 프로세스 흐름을 도식화 한 'Process structure chart' 이다.

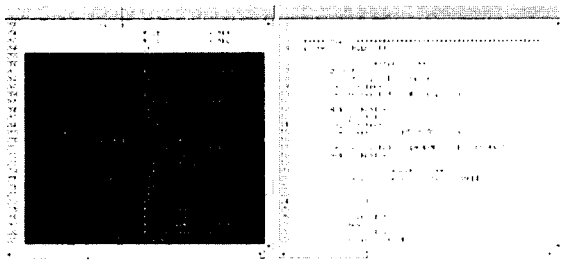


(그림 4) Call graph



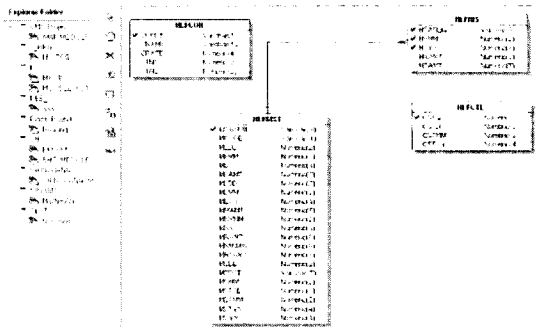
(그림 5) Process structure chart

2.2.2.2 Knowledge Mining



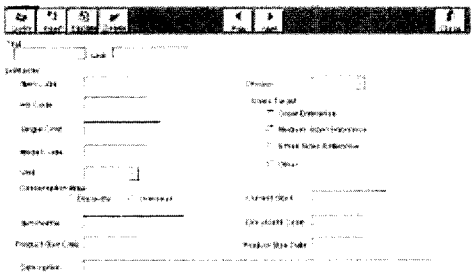
(그림 6) Business logic mining

위의 (그림 6)은 Business logic을 추출하는 화면이다. 왼편의 기존 소스코드에서 Business logic에 해당하는 부분을 찾아내어 컴포넌트화 시키고 컴포넌트에 대한 상세명세를 정의한다. 이 과정에서 동시에 소스코드의 변환이 이루어진다. 언어에 따라서 다른 문법 DB를 쓰고 기본적인 추출 및 변환 과정은 동일하다. 추출된 공통 컴포넌트 Repository에 저장하게 된다.



(그림 7) Database remodeling

위 (그림 7)은 Database의 remodeling을 하는 화면이다. Database의 스크립트 소스를 Import 하여 역공학에 의해 위와 같은 ERD(Entity Relation Diagram)을 만들어 낸다. 그리고 새로운 타겟 어플리케이션의 구조에 맞게 remodeling한다.

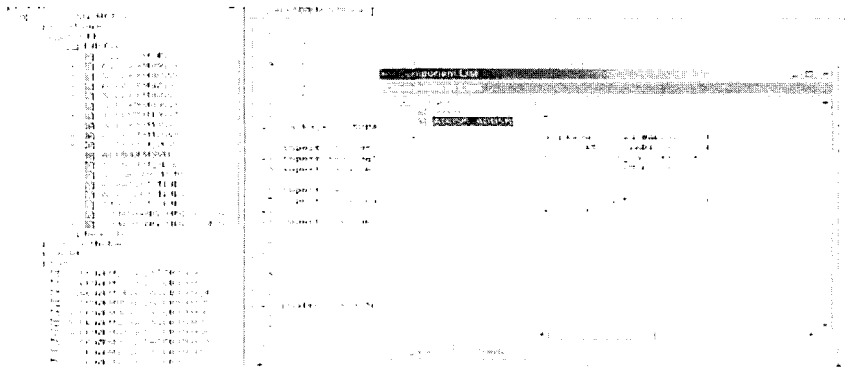


(그림 8) Screen Mining

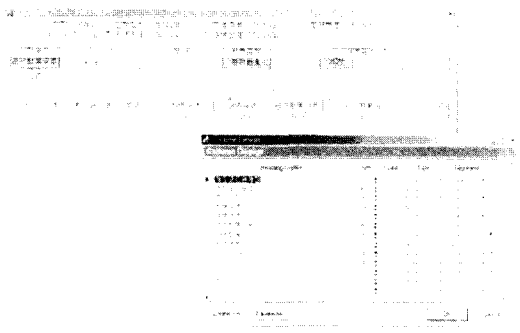
위 (그림 8)은 Legacy 소스코드를 re-engineering 하여 기존의 화면정보를 분석하는 화면이다. 위와 같이 4GL로 짜여진 화면소스코드를 Java의 소스코드로 변환하고 수정한다.

2.2.2.3 Component composing

(그림 9)는 Business Component를 composing하는 화면이다. 왼편에 보이는 Tree구조로 표현된 Component repository에서 기 제작된 Component와 추출한 Component를 조합



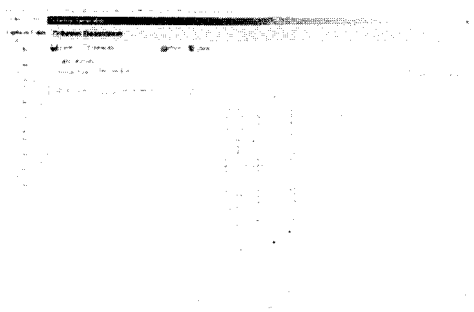
(그림 9) Business Component Composing



(그림 10) UI Component Composing

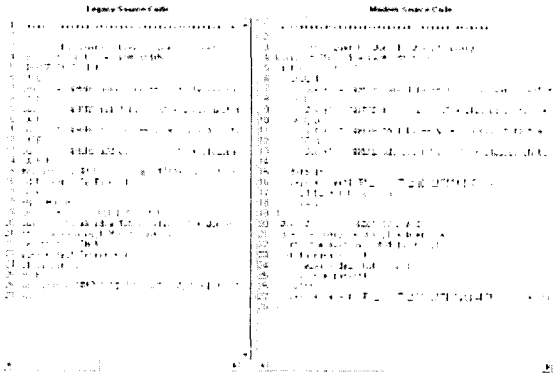
하여 Business Component를 구현한다.
위 (그림 10)은 UI component를 조합하는 화면이다. Screen Mining에서 추출한 화면과 Component Repository에 기 제작된 Component를 조합하여 Screen을 완성한다.

2.2.2.4 Transformation



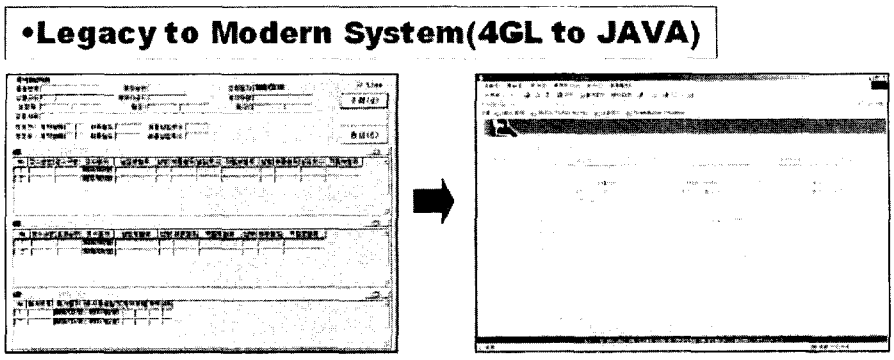
(그림 11) Database Transformation

위 (그림 11)은 Remodeling된 데이터베이스 Script를 생성하여 물리적인 Database를 구현하는 화면이다.



(그림 12) Source code Transformation

위 (그림 12)는 왼편의 Power Builder 소스코드에서 오른편의 Java소스코드로 변환하는 화면이다. 위와 같은 변환은 Legacy 프로그램 언어와 타겟 프로그램 언어의 문장구조를 Mapping할 수 있는 문법DB를 사용하여 변환한다. 또한 CBD환경의 Java프로그램에서는 필요 없는 소스코드를 제거하는 Code filtering기법을 적용한다.



(그림 13) 4GL 어플리케이션의 CBD Java 어플리케이션으로의 변환 결과

(그림 13)은 왼편의 4GL로 짜여진 Client Server 기반의 2-tier 어플리케이션이 오른편의 CBD기반의 웹서비스가 가능한 Java어플리케이션으로 변환된 화면이다.

2.2.3 변환의 핵심요소

앞에서 4GL 어플리케이션을 J2EE어플리케이션으로 변환하는 절차를 설명하였다.
이번에는 변환의 필수적인 요소들에 대해 알아보겠다.
첫째, 자동화된 도구를 사용한다.
자동화된 도구를 사용하지 않는다면 업무분석 시 많은 시간이 소요될 것이며, 빠른 개발라이프 사이클을 요구하는 시점에 많은 시간이 소요되며, 비용 또한 많이 발생한다.
그 보다 더 큰 문제점은 표준화 규칙을 적용하는데 있어서, 프로젝트 관리측면에서 자동화 도구를 사용하지 않으면 생산성이 떨어지며 중요한 요소가 누락되거나 프로그램 동기화에 문제점이 생길 가능성이 높다.
둘째, 변환도구를 사용하되 단순한 언어적인 변환만을 하였을 때, 기능, 아키텍처, 언어, 표현방법 등 복합적인 변환이 이루어 지지 않을 때에는 Web이나 CBD기반의 어플리케이션에서 가지고 있는 장점을 충분히 살리지 못한다는 점이 있다.

3 결론 및 향후 과제

프로그램언어와 데이터베이스의 변환에는 다양한 방법론이 제기되고 있다.
즉, EAI(Enterprise Application Integration) 방식, Wrapping방식 그리고 본 논문에서 다루고 있는 LM방법론(Legacy application to Modernization application)등이 이를 해결하고자 하는 방법론들로서 제기되고 있다.
이와 같이 소프트웨어공학 뿐만 아니라 소프트웨어산업에도 '재활용' 및 '재사용'에 대한 관심이 늘어가고 있다. 또한 이러한 추세에 발 맞추어 관련 기술이나 자동화 도구에 대한 개발에 대한 관심도 높아지고 있다. CBD기반의 어플리케이션을 구축하려면 무엇보다 소프트웨어 자원을 표준화하여 'Reusable'한 컴포넌트를 생성하는 것이 중요하다. 표준화 된 'Reusable'한 컴포넌트의 제작을 위해서는 그것이 Legacy에서 추출하는 경우이건 새롭게 구축하는 것이건 간에 표준화 된 도구를 활용하는 것이 효율적이다. 그러나 아직 많은 부분이 보완되어야 할 것이다. 예를 들어 기존 시스템에서 Business Rule을 추출하는 과정을 자동화 하기는 아직 부족한 부분이 많다. 소스코드를 기반정보로 하는 분석이기 때문에 다양한 소스코드에서 Business logic(Knowledge)를 자

동으로 찾아내기란 결코 쉬운 일은 아니기 때문이다. 앞으로 Business logic 추출의 자동화율을 높이기 위한 과제로, '관련 지식을 인공지능화 하여 Tool에 어떻게 적용할 것인가?' 하는 과제가 남아있다.

참고문헌

[1] CBDi, Component-Based Development Fundementals, 2002

[2] 박병형, 블록놀이와 CBD, 태영출판사, 2002

[3] 박병형, 'Web 환경시스템 구축 개발에 관한 연구', 연세대학교 산업대학원 최우수 논문

[4] John Bergey,Liam O' Brein, Dennis Smith, "Mining Existing Assets for Software Product Lines"

[5] Harry M. Sneed, "Risk involved in Reengineering Projects"

[6] Guido Dedene, " Object-Oriented COBOL: The Old, The Bad and The Ugly?"

[7] Harry M. Sneed, Dennis Smith, "Architecture and Functions of a Commercial Software Reengineering Workbench"

[8] Paul Asman, Federal Reserve Bank of New York, "Legacy Wrapping"

[9] Bulyonkov M.A, Fliatkina N.N, A.P Ershov Institute of Informatics Systems, Russian Academy of Sciences, " Exploring Dataflow in Legacy Systems"

[10] Kostas Kontogiannis, Johannes Martin, Kenny Wong, Richard Gregory, Hausi Muller, John Mylopoulos, "Software Evolution Through Program Transformations:An Experience Report"

[11] Andrey A.Terekhov, St.Petersburg State University.LANIT-TERCOM, "Automated extraction of classes from legacy systmes"

[12] Kathi Hogshead Davis, Northern Illinois University, "Combining a flexible Data Model and Phase Schema Translation in Data Model Reverse Engineering"

[13] Joe Hudicka, Alex Shterenburg, "The Complete Data Migration Methodology"

[14] Hajimu lida, Information Technology Center, Nara Institute of science and Technology, " Pattern-Oriented Approach to Software Process Evolution"

[15] Gerald Ebner, Hermann Kaindl, "Tracing All Around in Reengineering"

[16] Scott Tilley, "A Reverse Engineering Environment Framework"

저자약력

박 병 형

1975년~80년 삼성그룹 종합 전산실 (신세계 백화점) 대리, 군복무 전산화 (크레디트 카드 업무, 관리회계 업무)

1980년~81년 숭실대학교 전산실 연구원 (학사관리, 인사급여시스템, 도서관리)

1982년~89년 (주)유공(현재 SK) 정보시스템부 전문직 과장, 전산화(구매자제, SW QA, 개발센타장, 케이스툴개발)

1989년~ (주)케미스 대표이사 / CEO, 미국현지법인 NexxIT(회장)

2002년~호서대학교 벤쳐전문대학원 컴퓨터응용기술학과 박사과정

관심분야 : LM, CBD, ERP, CASE

이 메 일 : pph@camis.co.kr

양 해 술

1975년 홍익대학교 전기공학과 졸업(학사)

1878년 성균관대학교 정보처리학과 졸업(석사)

1991년 日本 오사카대학 기초공학부 정보공학과 S/W공학 전공(공학박사)

1975년~79년 육군중앙경리단 전산실 시스템분석장교

1980년~95년 강원대학교 전자계산학과 교수

1986년~87년 日本 오사카대학교 객원연구원

1994년~95년 한국정보처리학회 논문편집위원장

1995년~02년 한국S/W품질연구소 소장

2001년- 현재 한국정보처리학회 부회장

2003년- 현재미국ACIS학회 Vice President

1999년- 현재 호서대학교 벤처전문대학원 교수

관심분야 : 소프트웨어공학(특히, S/W 품질보증과 품질평가, 품질감리, 품질컨설팅, OOA/OOD/OOP, CASE, SI), S/W 프로젝트관리, CBD 개발방법론과 품질평가

이 메 일 : hsyang@office.hoseo.ac.kr