소프트웨어 원시코드로부터 모델링 정형명세(UML) 역공학 프로세스 연구

김 연 정*, 김병기**
*전남대학교 일반대학원 소프트웨어공학 협동과정
**전남대학교 전산학과
e-mail:peerkim@jeonju.ac.kr

A study of the reverse engineering for the modeling formed specification(UML) for from software source code

Yeon-Jung Kim
Dept. of Software Eng., Chonnam National University

요 약

모든 소프트웨어 시스템은 시스템 확장 및 아키텍처의 변화로 인한 잠재적 이주 필요성을 안고 있다. 그런데, 이 소프트웨어 이주 요구가 갖는 문제점은 새로운 시스템이 기존 시스템의 도메인 지식을 손상시키지 않도록 요청받는다는 점이다. 이로 인해 역공학의 필요성이 제기되나 하나의 소프트웨어 시스템은 다양한 측면의 시스템 요소를 갖고 있으므로 역공학을 통하여 직관적으로 이해가능한 설계정보를 유도해내는 것은 어려움에 직면한다. 따라서 본 연구에서는 하나의 레거시 시스템을 역공학 하기 위하여 필요한 프로세스를 제안하고, 역공학 결과물을 정형명세인 UML로 표현하는 사례를 통하여 검증하고자 한다.

1. 서론

1.1 연구배경

정보기술이 발전할수록 시장요구나 개발방법론의 변화는 점차 더 급속하게 이루어지고 있으며 소프트웨어 시스템의 생명주기는 더욱 단축된다. 해당 운용 환경 내부의 정보처리 요구량은 폭증하고 진보된 하드웨어 채용은 더욱 시급하다. 존재하는 소프트웨어 시스템은 변화하는 외부 환경에의 적응을 위한 확장 또는 새로운 시스템으로의 대체(replacement)에 직면한다. 이와 같이 모든 소프트웨어 시스템은 시스템 확장 및 아키텍처의 변화로 인한 잠재적 이주(移住, migration) 필요성을 안고 있다[1].

이주요구가 갖는 문제점은 새로운 시스템이 기존 시스템의 도메인 지식을 손상시키지 않도록 요구된다는 것이다. 실질적인 운용을 통하여 이미 확보되고 최적화된 문제영역지식(Domain Knowlege)는 반드시 재사용되어야 하기 때문이다. 그러나 재사용을 위해 필요한 현 시스템의 설계문서는 유지보수 기간 중에 훼손되거나 아예 유지보수되지 않는 경우가 많다. 따라서 현재 시점의 원시코드로부터 표준 모델링 언어로 표현된 정형명세를 추출해내는 역공학의 필요성이 제기된다.

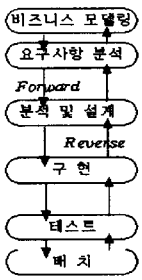
1.2 연구 목적과 방법

본 연구의 목적은 **원시코드를 기반으로 현재 시스템의 설계정보(UML)를 추출하는 역공학 프로세스를 제안**하는 것이다.

논문 구성은 다음과 같다. 2장에서 역공학과 정형명세에 관한 관련연구를 설명하고, 3장 본론에서는 제안한 역공학 프로세스를 전개한다. 이 과정은 **{원시언어 분석→ 데이터베이스 분석→ 원시코드 분석 → 도메인 전문가 조정 → 정형명세(UML)의 완성}**으로 사례연구와 함께 진행된다. 끝으로 4장에서는 제안된 프로세스로 유도한 정형명세의 성과와 문제점을 파악하고 향후 연구되어야 할 사항들을 제시한다.

2. 관련연구

2.1 소프트웨어 역공학의 소개



역공학(Reverse Engineering)이란 순공학(Forward Engineering)에 반대되는 개념으로서, **소프트웨어의 역공학이란 이전의(legacy) 또는 노후된 시스템의 이해를 용이하게 하기 위해 더 높은 추상화를 구성하는 과정**이다.

구조분석으로부터 비롯된 원시코드 추출 및 복원을 위한 노력은 많이 있어왔으나 역공학에 대한 보다 활발한 학계의 관심과 연구는 90년대 객체지향 개발방법론의 등장과 함께 시작되었다[1]. 특정 시스템 환경에서만 주로 진행되어오던 역공학과 재공학을 일반의 소프트웨어에도 적용하고자 하는 시도가 활발하게 일어나서, Cobol과 C 또는 Pascal과 같은 구조적 프로그래밍 언어로 기술된 프로그램을 C++, Java와 같은 객체지향적 언어로 변환하기 위한 다수의 방법이 제안되었다.

이후 컴포넌트 기반 개발방법론(CBD)[1] 등장에 발맞추어 컴포넌트의 추출 및 인터페이스의 분석을 위한 역공학 작업이 이루어지고 있으며, 향후에는 원시코드 변환이나 추출, 플랫폼 이전과 같은 국소적 접근보다도 비즈니스 룰 발견과 협력적 에이전트의 이전을 더 고려해야 할 것으로 예상된다[2].

시기	주안점	입력	주요도구	내용
초기	원시코드 분석	원시코드	역컴파일러 역어셈블러	정적 구조 분석 순차도 추출
90년대전	데이터베이스 역공학	Database 스키마	DB역공학	데이터 분석
90년대후	객체 분석	원시코드	매개언어	객체지향언어로 변환 객체추출 동적메시지 분석
2000년대	컴포넌트	분산된 소프트웨어 컴포넌트	컴포넌트 모델링	컴포넌트 분석 인터페이스 연결
미래	비즈니스 룰	협력환경	-	비즈니스 룰 발견, 협력에이전트 이전

표 2] 소프트웨어 역공학의 연구방향변화 및 주요 특징

2.2 소프트웨어 정형명세의 필요성

소프트웨어 개발은 현실세계의 문제 해결을 위한 면밀한 분석으로부터 시작한다. 실제 시스템 개발에는 분석가, 의뢰인, 프로그래머 등 다수의 참여자들이 의사 소통하여 원하는 소프트웨어 형상을 도출해낸다. 이렇게 요구사항을 하나의 형상으로 구체화하기 위해서는 오해의 여지가 없는 방법으로 소프트웨어 형상이 표현되어야 하지만, 상이한 이해 및 학습방법 및 선호도나 편리도 때문에 다양한 방법론과 표기법이 사용되어 왔다. 이로 인해 목표에서 어긋난 공통의 형상 도출을 위한 해결방법은 각 표기 방법으로부터 토론과 성취에 유용한 공통된 하나의 유형으로의 변환을 개발하는 것이다[3].

이와 같은 관점에서 전세계에서 공용으로 표기하고 이해할 수 있는 모델링 언어를 위한 표준화 작업이 진행되었다. UML(Unified Modeling Language)은 소프트웨어시스템 산출물의 명세 작성, 시각화, 구축, 문서화에 사용되는 표준 표기(notation) 시스템으로서 1997년 UML 컨소시엄에서 UML 버전 1.0 발표하였고 OMG[2]에 UML 1.1이 상정되어 표준 모델링 언어로 채택되었다[4].

UML 사용으로 표준화된 표기의 설계정보를 유지하고 현재의 시스템을 보다 직관적으로 이해할 수 있으며 비용절약적인 추가 순공학과 유사성 있는 새로운 시스템 구축에도 실제적인 유익을 줄 수 있다.

3. 본론

3.1 사례연구 대상

객체지향 분석 및 설계에서 가장 중요한 것은 '객체의 관점'에서 시스템을 분석하고 설계하는 것이다[5]. 본 연구에서는 사례연구를 위해 캠퍼스종합정보시스템의 '상담관리' 모듈의 내용을 설계하며 운용환경은 아래 표와 같다.

구분	제조회사	제품명
프로그래밍언어	Sybase	PowerBuilder 7.0.3
데이터베이스	IBM	Informix 9.0
모델링 도구	IBM	Rational Rose 2002

표 3] 사례 연구의 구현 환경

3.2 역공학의 추출

역공학의 현재에서 원시코드로부터의 완벽한 상위단계 문서 추출은 불가능하며, 완벽한 자동화도구 설계는 더더욱 불가능하다. 이를 보완하기 위하여 역공학 통합환경이 논의되며, 본 연구에서도 전체적인 "추출 도구와 전문가(domain expert) 조정, 명세도구"라는 환경을 구성하였다.

1) 컴포넌트 기반 개발(Component-Based Development) : 서로 다른 플랫폼에 분산된 이질적(heterogeneous)인 소프트웨어 컴포넌트들을 효과적으로 분류, 유통 및 조립하는 기술 필요성이 제기됨에 따라 등장한 컴포넌트 생산, 선택, 평가 및 통합으로 구성되는 소프트웨어 개발 방법의 가장 새로운 패러다임.

2) 객체 관리 그룹 (OMG : Object management Group)
1989년에 발족된 객체 지향 기술의 보급과 표준화를 추진하는 비영리 단체. 객체 지향 기술 분야에서 업계 표준인 객체 관리 구조(OMA)에 준거하여 서로 다른 기종의 컴퓨터 간의 조작 환경을 통일하는 것을 목적으로 한다.

도구	자동화도구	입력 자료	생성 자료
추출 도구	Rational Rose/PB	원시 코드	캡슐화된객체
(시스템분석기)	-	데이터베이스	데이터 사전
전문가 조정	-	구조화된메모	조정설계자료
명세 도구	Rational Rose	모든추출자료	UML

표 4] 역공학 통합환경에 따른 사례 구현 도구

아래는 역공학전개를 통해서 유도해야 할 다이어그램 목록과 그 설명이다.

종류	설명
클래스	시스템에서 사용되는 객체의 틀 표현
객체	클래스의 인스턴스 표현
유스케이스	사용자의 입장에서 본 시스템의 행동
상태	시간에 따라 달라지는 객체의 상태
시퀀스	객체들끼리 주고받는 메시지의 순서 표현
활동	유스케이스/객체의 동작중에 발생하는 활동
협력	시스템 요소간의 협력관계
컴포넌트	시스템을 이루는 물리적 요소 및 그 관계
배치	컴퓨터를 기반으로 한 시스템의 물리적 구조

표 3] UML로 구현해야 할 기본 9가지 다이어그램

역공학 프로세스를 위한 활동(activity)은 소프트웨어 시스템 전체 환경을 고려하여 다음 표 6]과 같이 제안하며 세부 내용은 아래와 같다.

작업	입력 자료	주요 활동	
		자동화도구	생성 자료(UML자료)
		사례연구에서의 처리방법	
원시 코드 정제	원시 코드	불필요한 코드 삭제 및 단순한 구문 최적화	
		-	정제된 원시코드
		프로그래머의 코드 단순화 작업	
프로그래밍 언어의 분석	언어의 설계 정보	프로그래밍 언어에 대한 문법 지식정보 수집	
		Rose/PB	프로그래밍언어의 **문법해석사전**
		파워빌더의 문법지식을 Addin module이 보유	
DB 분석	DB 스키마	데이터베이스 테이블 →대단위 클래스로 분류 테이블의 칼럼 → 클래스의 속성으로 분류	
		Powerbuilder	데이터베이스의 **테이블사전** 불완전한 **클래스다이어그램**
		테이블 생성시 기록한 PB 확장속성 테이블	
원시 코드 정적 분석	원시 코드	캡슐화된 객체의 표면적인 작업 분석 윈도우 객체 → UML stereotype class 연결	
		Rose/PB	**컴포넌트 다이어그램** **패키지,클래스 다이어그램**
		윈도우 객체 → UML stereotype class 연결 캡슐화된 객체의 표면적인 구성 분석	
	분류된 클래스	객체분석을 통해 추측 가능한 Use Case 구성	
		Rose/PB	**유즈케이스,객체 다이어그램**
		윈도우(window, library)→ UseCase에 연결	
원시 코드 동적 분석	UI	User Interface 설계정보 추출, 객체지향언어에서 제공하는 객체 용도, 이벤트, 속성 분석, UI분석을 통해 추측 가능한 상태도 구성	
		PB_UI	**시퀀스,상태 다이어그램**
		PB에서 제공하는 control들의 메타사전 캡슐 내부의 작업들이 구체적으로 분석됨	
전문가 조정	취합된 정보	유스케이스의 의미적 재분류 외부 액터의 배치	
		Rose	**시퀀스,상태 다이어그램** **패키지,배치 다이어그램**

표 6] 제안된 역공학 프로세스에 따른 활동과 도구, 결과물

가) 원시코드 정제

불필요한 코드 삭제 및 단순한 구문 최적화 작업이다. 정제된 원시코드를 만들기 위해 프로그래머가 직접 코드 단순화 작업을 수행한다.

나) 프로그래밍 언어 분석

프로그래밍 언어의 문법 해석 사전을 구현해두는 작업이다. 프로그래밍 언어의 문법 지식정보를 수집하기 위한 가장 기본적인 분석 지식 구축 단계로서, 프로그래밍 언어의 키워드, 데이터타입, 연산자, 구문 등을 이후 작업에서 연결할 수 있도록 설계한다.

사례에서는 원시코드 프로그래밍 언어의 기본 분석을 위해서 파워빌더를 위한 애드인(Addin)모듈을 사용하였다. 이 모듈은 메텍스(Metex)사에서 개발하였으며 5.0, 6.0, 6.5, 7.0 까지 지원이 가능하다[6].

다) 데이터베이스 분석

데이터베이스의 설계자료를 분석하여 데이터베이스 사전과 테이블에 저장되는 정보로 개략적 클래스 다이어그램을 구현하는 작업이다. 응용프로그램의 명세를 복구하는 것은 첫째로 그들의 자료(Data)를 복구하는 것을 필요로 한다[7]. 최소한의 요구사항 수집은 데이터베이스 분석만으로도 가능할 정도로 중요한 과정이며, DBRE(Database Reverse Engineering)는 독립적인 역공학 분야로서 업체별로 상용 툴도 다수 제공되고 있다.

사례에서는 테이블 생성시 기록해 둔 파워빌더의 확장속성 시스템 테이블을 이용하여 테이블 사전을 구현하고, 각 테이블을 하나의 클래스 및 속성으로 분류하였다.

테이블	저장되는 확장속성정보
PBCatCol	header, label등과 같은 컬럼데이터 및위치
PBCatEdt	Edit Style의 이름과 정의
PBCatFmt	Display format 이름과 정의
PBCatTbl	dlfma, 폰트, 주석과 같은 테이블 데이터
PBCatVld	Validation rule의 이름과 정의

표 7] 파워빌더에서 유지하는 확장 속성 시스템 테이블

라) 원시코드 정적 분석

윈도우 객체별 분석 원시코드 파일로부터 윈도우 객체의 구조를 분석해내는 작업이다. 언어별로 다르게 제공하는 윈도우 객체들은 UML의 스테레오타입 클래스로 정의하여 분류된다.

사례에서는 원시 파일(pbl)을 입력하면 모든 윈도우 객체들이 자동화 도구에 의해 역공학된다. 별도 언어사전을 구축한 경우에는 파워빌더에서 텍스트 형식으로 export되는 원시코드 파일들(srw, srd, sra, srs, sru, srj)을 입력으로 사용할 수 있다.

하나의 윈도우는 하나의 클래스로 캡슐화된 구조가 분석되며 속성변수, 이벤트나열 및 객체 클래스 연결이 이루어진다. 윈도우 객체마다에 연결된 스크립

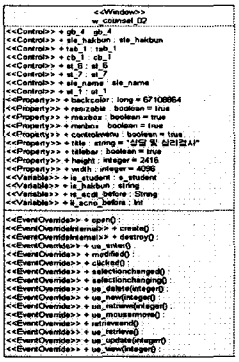


그림 2] 윈도우 객체 중 하나의 윈도우가 역공학된 결과

트는 별도문서(document) 형식으로 첨부된다.

유즈케이스 분석 객체 분석을 통해 추측 가능한 유스케이스를 구성해내는 작업이다. 현재 원시코드로부터 유즈케이스 개념을 자동화 도구를 통해 추출해내기는 매우 어렵다. 그러나 일반적인 GUI 인터페이스를 사용하는 프로그램에서는 하나의 윈도우가 하나의 주요 활동을 처리하며, 하나의 실행파일이 큰 하나의 유즈케이스를 표현한다고 가정한다.

본 사례에서는 파워빌더에서 하나의 라이브러리 파일이 윈도우를 하나의 상위 유즈케이스가 되고, 윈도우별로 하나의 하위 유즈케이스를 가지며, 기능성 버튼별로 그 하위 유즈케이스를 가지는 것으로 가정하여 분류하였다.

마) 원시코드 동적 분석

사용자 인터페이스 분석(UIRE)

사용자 인터페이스를 분석함으로써 동적 다이어그램을 추출해내는 작업이다. 언어마다 인터페이스 방식이나 이벤트 처리 기능이 다르므로 사용자 인터페이스 설계정보 추출을 위해서는 프로그래밍 언어의 이해구조 중 컨트롤(객체)의 용도, 이벤트간의 순서 분석을 통해 상태 다이어그램을 유도한다.

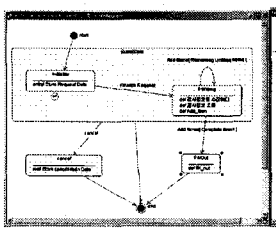


그림 3] 상태 다이어그램

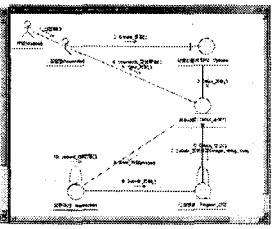


그림 4] 협력 다이어그램

바) 전문가 조정

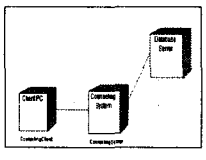


그림 5] 배치다이어그램

유즈케이스와 액터의 연결이나 소프트웨어 원시코드에 포함되지 못하는 시스템의 물리적 배치 다이어그램은 전문가 조정이 없이는 추측해내기 어렵다. 이전단계까지의 정보를 상향 취합하여 조정하며 최종 다이어그램 구성을 결정한다.

4. 결론

4.1 연구 결과 분석

본 연구에서는 소프트웨어 시스템 환경을 전체적으로 분석하여 통합 역공학 프로세스를 전개하였다. UML은 매우 다양한 뷰(view)를 제공하는 모델링 언어이므로 기본적인 다이어그램의 추출을 위해서는 소프트웨어 형상에 다각적으로 접근해야 한다. 이를 위해 다양한 역공학 입력자료를 정의하고 UML 다이어그램으로 연결한 결과, 보다 정확한 객체추출과 클래스 다이어그램 유도가 이루어졌다.

4.2 향후 연구방향

원시 언어 분석을 통한 역공학은 성과를 보고한다. 그러나 여전히 동적관점을 지원하는 다이어그램의 구현은 미흡한 점이 있다. 그러나 객체지향 언어로 발전해갈수록 언어 자체가 프로그래밍 방식을 유도하게 되어 있어 역공학의 성과를 높이는 것으로 나타나므로 이를 위해 이벤트 패턴의 분석과 사용자 행위추측을 통한 협력 다이어그램 유도에 관한 지속적 연구가 이루어져야 할 것이다.

[참고문헌]

[1] Gannod,G.C. "Using informal and formal techniques for the reverse engineering of C programs ". 1996. WCRE

[2] Hausi A.Miller, Kenny Wong Margaret-Anne Storey,' Reverse Engineering Research should TargetCooperativeinformationSystem Requirements'. Wcre '98

[3] Shari Lawrence Pfleeger. Software Engineering 'theary and practice, Prentice-Hall

[4] Joseph Schumuller, "UML 객체지향 설계 second edition". 인포북

[5] 조완수, "UML 객체지향 분석 설계". 홍릉출판

[6]http://www.metex.com/metex.asp?link=serviceCS&file=htdocs/services-product/clientserver/softwaretools/rosepowerbuilderlink.html

[7] hainaut, J-L "Requirements for information system reverse engineering support". IEEE 1995

[8] Lutsky,P. "Automating testing by reverse engineering of software docunmentation".IEEE 1995