

Adams-H 를 위한 문서화 시스템 설계 및 구현

이승훈*, 조현훈**, 류성열***

*숭실대학교 컴퓨터학과

{shlee, hhcho}@selab.ssu.ac.kr syrnew@computing.soongsil.ac.kr

A Design and Implementation of Documentation System for Adams-H

Seung-Hun Lee* , Hyun-Hoon Cho**, Sung-Yul Rhew***

*School of Computing, Soongsil University

요 약

기존의 2 Tier 나 3 Tier 형태의 소프트웨어 개발 시 생산성과 효율성을 향상시키기 위하여 RAD Tool 을 많이 사용하였다. 이러한 RAD Tool 의 사용은 생산성과 효율성을 향상 시켰지만 소프트웨어의 유지보수나 성능 개선을 위한 작업 시 코드의 중복이나 복잡성 등의 증가 등 많은 문제점들을 발생 되었으며, 이는 소프트웨어의 유지보수에 많은 문제점을 가져왔다.

이러한 문제점들은 역공학 기법을 통해서 해결이 가능하다. 역공학 기법을 이용하여 소프트웨어의 구조를 이해할 수 있는 정보를 추출하고 추출된 정보를 체계적으로 관리할 수 있는 시스템을 구축 하여 문서화 하는 것이다. 이에 본 논문에서는 소프트웨어의 소스코드를 분석하여 시스템 변경 등 의 작업 시 문서화를 지원하여 생산성과 효율성을 증가시킬 수 있는 시스템을 설계 구현하였다.

1. 서론

잘 작성된 소프트웨어라도 사용자의 요구사항이나 운영 환경의 변화에 의해 지속적인 유지보수나 시스템의 변화가 발생 한다. 그러나 이러한 작업을 위해서는 프로그램을 분석하고 이해하는 과정이 반드시 필요하다. 이는 작업 시간과 비용 절감의 측면도 있지만 궁극적인 목적은 전체적인 시스템의 안정성을 유지하기 위함이다.

프로그램의 분석과 이해는 개발 시 작성된 문서들을 참조하는 것이 가장 일반적인 방법이다. 그러나 수년 동안 비 구조화되고 비 문서화된 문서화 방법으로 인해 대부분의 소프트웨어 개발 문서가 없어졌거나, 문서가 있다 하더라도 수정이 많이 가해져서 프로그램의 구성 요소와 그들 간의 관계를 식별하지 못하는 것이 현실이다.

본 논문에서는 대상 소프트웨어 시스템의 원시코드를 분석하여 소프트웨어 구성 요소들과 그들간의 관계를 추출하여 가장 하위 수준의 추상화인 코드로부터 상위 수준의 추상화 결과인 분석 또는 설계 정보를 문서화로 만들어 내는 시스템을 설계 및 구현하였

다[5][7].

이렇게 작성된 문서화 산출물들은 소프트웨어 유지보수성의 향상과 생명주기의 확장 및 컴포넌트화를 위한 기초 자료로 사용될 수 있을 것이다[6].

본 논문의 구성은 다음과 같다. 2 장에서는 시스템의 이해를 위해 기본 개념을 기술하고, 3 장에서는 시스템을 설계하며, 4 장에서는 구현하고, 마지막으로 5 장에서는 결론 및 향후 연구과제를 제시한다.[1]

2. 관련 연구

2.1 문서화(Documentation)

문서화란 사용자가 요구하는 소프트웨어를 개발하기 위하여 적용하는 개발 프로세스에서 정의된 산출물들을 각각의 개발 단계마다 작성하는 것을 말한다 [1][4].

다음 표 1 은 개발 단계에서 만들어지는 문서화 목록을 정리한 것이다.

표 1. 개발 단계별 문서화 목록

개발 단계	문서화 목록
요구분석	요구분석서
기본설계	기본설계서
상세설계	상세설계서
구현	프로그램작성계획서
시험	시험결과서
검수	검수의뢰서
프로젝트완료	프로젝트완료보고서
유지보수	유지보수계획서

위에 정리된 문서화 목록들은 개별적으로도 중요하지만, 다음 개발 단계 작업에 중요한 자료로 활용된다[3].

2.2. RAD Tool 의 개발 단계

본 논문은 RAD(Rapid Application Development) Tool 중의 하나인 Visual Basic 으로 작성된 Adams-H 소프트웨어를 대상으로 하고 있다. 사용자에게 보여지는 UI와 사용자들의 Event 에 대한 동작을 처리하는 Procedure 들로 구성이 되는 폼파일과 프로젝트의 시작과, 일반적인 Business Logic 들을 처리하는 공통 코드나 함수들의 집합으로 구성 되어지는 모듈파일, 사용자 정의 자료형의 일종으로 속성과 메소드를 구성 되어지는 클래스파일을 사용하여 소프트웨어를 개발하게 된다[2][8].

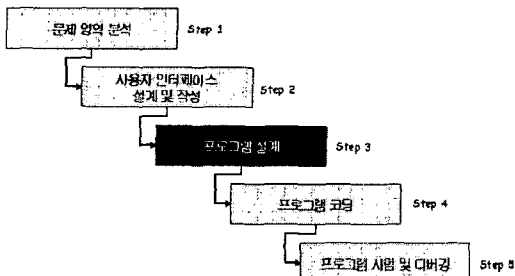


그림 1. RAD Tool 의 개발 단계

Visual Basic 은 위와 그림 1 과 같은 일반적인 RAD Tool 의 5 단계의 개발 단계를 따른다. Step 1 은 문제 영역의 분석 및 개발과 관련된 제반 사항들을 결정하게 된다. Step 2 는 Step 1 에서 분석된 자료를 바탕으로 사용자 UI 에 대한 기초 설계와 Visual Basic 을 사용한 작성을 하게 된다. Step 3 은 작성된 사용자 UI 에 대하여 Pseudo Code 를 작성한다. Step 4 는 Pseudo Code 를 Real Code 로 바꾸는 작업을 한다. 마지막으로 Step 5 에서는 작성된 전체 코드를 시험 및 디버깅하여 숨어 있는 문제점과 안전성을 높이는 작업을 하게 된다.

3. Adams-H 문서화 시스템 설계

3.1. 범위

본 시스템 2.2. RAD Tool 의 개발 단 중 Step 3 프로그램 설계 부분에서 작성되어야 하는 산출물인 프로그램 작성 계획서의 역 문서화 기능을 제공한다. 따라서 Step3 을 구성하는 File 과 Procedure 구성 요소들을 추출하여 문서화와 각종 검색 기능을 제공한다[9].

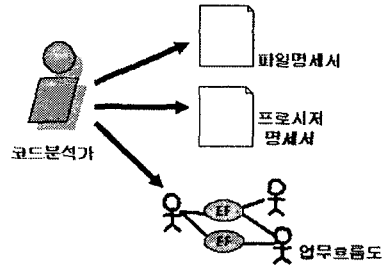


그림 2. Step 3 산출물의 내용

위의 그림 2 에서와 같이 소프트웨어 초기 개발 단계에서는 코드 분석가가 설계 단계에서 파일명세서와 프로시저명세서, 업무흐름도를 산출물로 작성하여 개발자가 시스템 개발 시 활용하도록 한다. 이 중 파일명세서에는 VB 의 폼, 모듈, 클래스 파일에 대한 세부 내용들이 작성 된다. 다음으로 프로시저 명세서에는 각각의 파일들이 수행하게 되는 Business Logic 을 포함한 코드들에 대한 세부 내용들이 포함된다.

결국 본 시스템은 이러한 코드 분석가의 역할을 코드 분석기가 대신해서 산출물을 코드에서 역으로 만들어주는 기능을 수행하게 된다.

3.2. 구성

3.1 의 범위에서 언급한 바와 같이 본 시스템은 코드 분석가가 개발자를 위해 작성했던 산출물을 제공하기 위해 그림 3 과 같이 구성되게 된다.

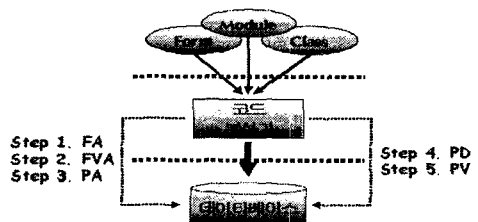


그림 3. 시스템 구성도

코드 분석기는 소프트웨어를 구성하고 있는 코드 파일에서 먼저 FA(File Analysis) 정보를 생성한다. FA 는 Visual Basic 의 3 가지 File Type 에 대한 기본적인 정보들로 문서화를 기능의 가장 핵심 정보로 사용된다.

FA 정보가 작성되면 코드 분석기는 File 단위의

Variable 정보를 작성하는 FVA(File Variable Analysis) 정보를 생성한다. 다음으로 코드 분석기는 PA(Procedure Analysis) 정보를 생성하는데. 이 정보는 FA 에 포함되어 있는 Procedure 들에 대한 기본 정보로 이들 PA 에 대한 상세 정보는 코드 분석기의 PD(Procedure Description)정보 생성에서 만들어진다.

생성된 PD 정보에는 내부에서 사용되는 Variable 이 존재하며, 이 정보는 코드 분석기의 PV(Procedure Variable)를 통해서 생성된다.

이렇게 생성된 정보들은 그림 4 와 같이 두 가지 산출물로 작성이 되어 사용자에게 제공되거나 시스템의 검색 기능을 통하여 검색을 할 수 있는 기능을 제공한다.

그림 4. File 명세서와 Procedure 명세서 유형

3.3. 기대효과

본 문서화 시스템은 그림 5 와 같이 유지보수자에게는 소프트웨어의 Customizing, Restructuring 을 할 때 UI 부터 해당 기능의 Biz-Logic 을 문서로 추적하여 수정 및 관리할 수 있는 지침서로, 신입사원에게는 개발된 소프트웨어의 UI 구조 및 프로그램에 대한 이해와 프로그램에서의 업무흐름이해 및 코드 분석을 할 수 있는 자료로 활용이 가능하다.

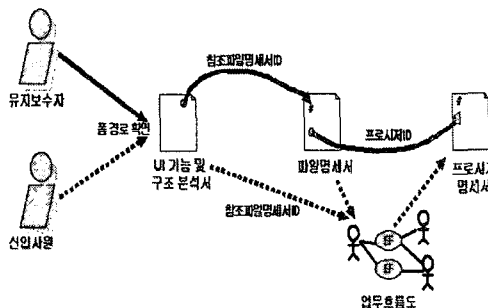


그림 5. 산출물간의 연관 관계

4. Adams-H 문서화 시스템 구현

본 시스템의 구현은 Windows 환경 하에서 Visual Basic 6.0 개발 언어와 SQL Server 와 Access 2000 을 사

용하여 개발 하였다. 코드 분석기는 대상 파일들(폼, 모듈, 클래스)에서 항목 추출 기능을 통해 정보를 추출하여, DB Server 에 기록한다.

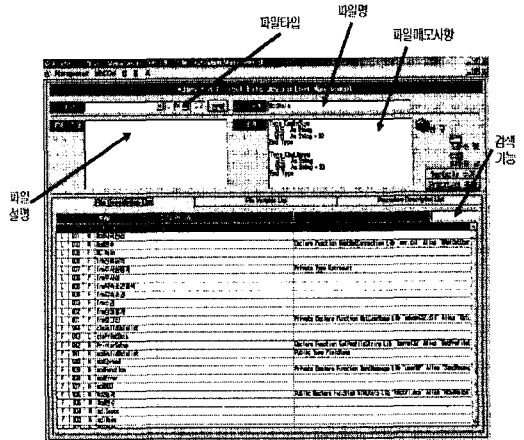


그림 6. FA, FVA, PA 정보 생성 및 검색 창

File Analysis, File Variable Analysis, Procedure Analysis 정보들은 그림 6 에서와 같이 다양한 조건으로 검색을 할 수도 있으며, 산출물 필요 시 출력 기능을 이용하여 File 명세서와 Procedure 명세서를 출력할 수 있다.

그림 7 은 Procedure Description, Procedure Variable 정보를 검색하는 창으로 File Analysis 정보를 통해 생성된 정보들로 상호 참조가 가능하고, 시스템 유지보수 작업 시 의사 결정에 중요한 자료로 사용할 수 있다.

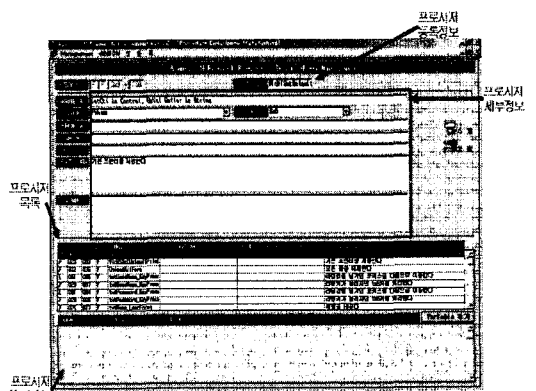


그림 7. PD, PV 정보 생성 및 검색 창

5. 결론 및 향후 연구

RAD Tool 을 사용한 개발은 편리성과 기간의 단축이라는 이점을 제공한다. 그러나 이러한 이점은 유지보수의 문제와 기간의 단축에 의한 문서화의 생략 및 부실한 문서화 문제들을 발생시켰다.

본 논문에서는 역공학 기반의 문서화 시스템 구현을 통해 이러한 문제들을 해결해 보았다. 또한 소프트웨어를 통해 문서화 정보를 생성하기 때문에, 최종 정

보를 유지할 수 있는 이점까지도 얻을 수 있었다.

향후 연구되어야 할 부분으로는 정보 생성시 레퍼지토리를 구축하는 것으로 이는 재사용 가능 모듈을 검색할 수 있고, 필요로 하는 코드의 재구성과 역공학과 재공학을 통하여 재사용 가능한 컴포넌트 모델 추출 기능을 제공할 수 있을 것으로 생각된다[6].

참고문헌

- [1] Roger S. Pressman. *Software Engineering, A Practitioner's Approach*, 3rd Ed. McGraw Hill, 1997.
- [2] Alka R. Harriger *Computer Programming with Visual Basic 6 A Problem-Solving Approach*, Que E&t, 1999.
- [3] Thomas M. Pigoski, *Practical Software Maintenance*, Wiley, pp. 37~50, 1997.
- [4] N.T. Fletton and M. Munro, *Redocumentating S/W Systems using Hypertext Technology*, IEEE Intl conf. Software Maintenance, 1988.
- [5] H. Mili, F. Mili, and A. Mili, *Reusing Software : Issues and Research Directions*, *IEEE Transactions on Software Engineering*, Vol. 21, No 6, pp. 528-561, June 1995.
- [6] ISO 12207, "ISO/IEC Standard for Information Technology - Software life cycle process", ISO/IEC/JTC/SC7, March, 1998.
- [7] McGill, R., *Reverse Engineering and Reengineering - not yet?*, In *Software Reuse and Reverse Engineering in Practice*, Hall, P.A.V, Chapman&Hall, 1992.
- [8] 이현기, 신규상, 역공학에서 재문서화를 지원하는 도구, 정보처리학회 춘계 학술발표논문집, 제 5 권 제 2 호, pp. 433-436, 1998.
- [9] 이종호, 류성열, "객체지향 기반의 Refactoring 프로세스", 한국정보과학회 논문지, 제 7 권 4 호, pp.299~308, 2001.8.