

기능 분석 그래프에 기반한 요구 사항 분석 및 테스트 경로 검증 방법

이지현, 김진삼
한국전자통신연구원, S/W 공학연구팀
e-mail : {jihyun, jinsam}@etri.re.kr

A Study on Requirement Analysis and Test Path Verification through Function Analyzing Graph

Jihyun Lee, Jin-Sam Kim
Electronics and Telecommunications Research Institute, S/W Engineering Research
Team

요 약

제품에 대한 다양한 사용자의 요구 사항을 지원하기 위해 제품 개발 프로세스는 초기 단계인 분석 및 설계 단계에서부터 다양성을 지원할 수 있도록 개발 주기가 이뤄지고 있고 이러한 추세는 차별화된 서비스의 소프트웨어를 내장한 시스템(이하 임베디드 시스템)의 경우 보다 활발히 이뤄지는 경향이다. 요구 사항에서 공통점은 시스템 개발에 있어 재사용성을 높일 수 있는 중요한 부분이므로 요구 사항에 대한 다양성을 분석하고 표현하여 다양한 제품이 개발이 개발 초기 단계에서부터 이뤄 질 수 있도록 하는 것이 본 논문의 목적이다. 따라서 분석의 기반이 되는 기능 표현 방법은 중요하며 설계와 개발 단계로 연결되기 위해 분석된 기능에 대한 검증이 이뤄져야 한다.

1. 서론

최근 시스템에 대한 개발은 사용자의 다양화된 요구 사항에 따라 다른 크기, 기능, 설계, 성능의 제품으로 나오고 있다. 시기 적절하게도 최근 임베디드 시스템의 개발에 있어서 다양화된 제품을 개발하는 것은 강한 경향으로 나타나고 있다. 단일 제품을 개발하는 것과 다양한 제품을 개발하는 것은 일반적으로 제품의 개수만큼 개발 비용이 소요된다고 생각할 수 있다. 그러나 제품 개발에 있어 현실적인 개발 환경은 시간과 개발 인력에 있어 지속적으로 부족한 상태였고 이러한 현상은 더욱 심해질 것으로 예상된다. 이러한 상황에 있어 경쟁력 있게 제품을 개발하는 방법은 개발에 필요한 모든 구성 요소를 재사용성이 높도록 설계하고 개발하여 필요할 때 사용하기 좋게 관리하며 총체적인 개발 프로세스가 효율적으로 수행되도록 하는 방법이다. 이러한 시스템 개발에 대한 총체적인 체계는 개발 방법론으로 표현되고 있고 그 중

컴포넌트 개발 방법론이 1990년대부터 소개되고 적용되어 왔다.

컴포넌트에 기반한 시스템 개발에 있어 기존의 데스크톱 시스템 개발은 재사용성을 높이기 위한 수단으로 개발 단위를 컴포넌트화하고 구현된 컴포넌트들을 서로 결합하여 전체 시스템에서 제공하고자 하는 기능을 제공한다. 컴포넌트의 재사용이 다양화된 요구 사항을 지원하기 위해서는 컴포넌트에 대한 개념이 사용자의 다양한 요구 사항을 반영할 수 있도록 설계되어야 한다. 따라서 컴포넌트가 재사용 단위가 되면서도 요구사항의 다양성을 컴포넌트의 특정 특징으로 반영하고 변경할 수 있는 방법을 제공해야 한다.

본 논문에서는 사용자의 요구 사항을 기능으로 표현하고 테스트해볼 수 있는 방법을 제공한다. 본 논문의 구성은 다음과 같다. 제 2장에서 선행 연구의 동향과 개선 방향을 살펴 보고 제 3장에서 시스템 공학 측면에 있어 요구 사항 표현 범위를 결정지어

설명한다. 제 4 장에서 요구 사항을 표현하는 방법에 대해 제안한다. 제 5 장은 요구 사항 분석 및 검증에 대한 평가해 보고 제 6 장에서 결론을 맺는다.

2. 선행 연구

임베디드 시스템은 시스템 구성 소프트웨어와 하드웨어의 구성 종류를 달리하며 유사한 기능과 차별화된 서로 다른 서비스를 제공하는 시스템으로 개발되고 있는 상황이다. 시스템을 개발하기 지원 도구 연구는 소프트웨어 공학에서 제공되는 CASE(Computer-Aided Software Engineering) 도구는 모델링, 컴파일링, 디버깅의 기능을 지원하는 도구가 대부분이었다. 이중 모델링 도구의 경우 시스템에 대한 계층을 서브 시스템, 컴포넌트, 클래스, 함수의 수준으로 분석하는 도구가 많은 비중을 차지해 왔다. Holmes[1]는 유사한 기능을 갖고 서로 다른 제품들(이하 제품군)을 개발하기 위해 상호 의존하는 활동을 지원할 수 있는 도구의 지원이 중요하므로 제품군의 개발을 위해 외부 도구와 연계할 느슨한 형태의 하부 구조를 제안하고 기존의 도구에서 발생하는 데이터를 관리하고 통합하는 방식을 제안한다. 이는 기존의 방식으로 요구 사항을 데이터 모델로 표현하고 통합된 도구를 통한 제품 생성 단계로 연계됨을 제안한다.

Together[2]와 같은 개발 통합 도구는 설계, 개발, 배포와 같은 기능을 통합적으로 제공하지만 모델링은 UML 을 지원하는 편집기 기반의 도구로 제공되어 UML 의 다이어그램을 시스템 설계자가 표현하도록 한다. 이는 이해 당사자들한테 받은 요구 사항을 표현하고 분석하기 위한 단계를 별도로 수행하고 설계, 구현, 배포로 이어지는 작업들이 이뤄지도록 지원하기 때문에 제품군에 대한 분석이 설계와 자연스럽게 연결되기 어렵고 제품군의 요구 사항을 분석하기 위한 기능을 제공하지 못한다. 기존에 시판되고 있는 유명한 도구는 제품군에 대한 분석이 UML 에 한정되어 있어 요구 사항 도출 단계에 알맞게 유연하면서도 안정된 도구를 찾기 어려운 상태이다.

제품군의 요구 사항을 표현하고 표현된 요구 사항에 대한 다양성을 도출하기 위해서 요구 사항에 대한 검증으로 이어질 수 있고 사용하기 쉬운 도구가 필요하다. 제품 정보의 분석 방법[3][4] 연구에서는 UML 에 기반한 기존의 모델링 도구와 별도의 데이터베이스를 이용하여 제품에 대한 다양한 분석 정보를 저장하고 관리할 수 있도록 시각적인 분석기를 이용하여 요구사항의 저장과 질의를 가능하게 하는 기능을 갖는다. 하지만 이는 제품의 분석 대상을 시스템, 함수, 품질 데이터에 대한 분석만을 가능하게 할 뿐 의미적인 기능 분석과 제품군의 다양성이나 특성을 고려하는 기능이 배제되어 있어 제품 군의 생산을 위한 제품 분석 정보를 도출하기 위한 사용으로는 부족함이 있다.

3. 요구 사항 표현 범위

요구 사항 분석 단계는 제품 개발의 상위 수준에서

이뤄지므로 제품 생산에 관련된 참여자들이 이해를 같이 하기 위해 특정 모델링 언어를 알아야 하거나 특정 도구를 사용할 수 있어야 한다는 제약 사항 없이 제품에 대한 요구 사항을 간편하게 표현할 수 있어야 한다. 따라서 상위 수준의 요구 사항에 대한 표현 분석 및 검토가 효과적으로 이루어 질 수 있다.

또한 소프트웨어 개발뿐만 아니라 하드웨어와 소프트웨어를 모두 포함하는 시스템을 개발할 수 있도록 개발 참여자들이 알고 있는 수준의 하드웨어와 소프트웨어가 기능으로 식별되어 표현될 수 있어야 한다. 본 장에서 포함하는 내용은 시스템 공학 프로세스(System Engineering Process: SEP)[5]의 8 개의 서브 프로세스의 초기 4 가지 프로세스인 요구 사항 분석, 요구 사항 검토, 기능 분석, 기능 검토를 포함한다.

<표 1> 시스템 공학 프로세스의 요구 사항 분석 관련 서브 프로세스 기능

| 서브 프로세스 | 설명 |
|----------|--|
| 요구 사항 분석 | 사용자로부터 도출된 시장 요구, 시스템의 요구사항, 제약사항으로부터 사용자가 기대하는 내용, 프로젝트 수행상 제약 사항, 기능/성능 요구 사항, 기능 및 디자인 특성을 식별하고 정의한다. |
| 요구 사항 검토 | 요구 사항 분석 프로세스에서 정의된 사용자의 기대를 도출된 요구 사항이 만족시킬 수 있는 지를 제약사항에 기반해 검토하고 충돌되는 내용이 있는 지 확인한다. |
| 기능 분석 | 요구 사항에 정의된 문제를 기술하고 시스템 디자인 구성 요소로 만족될 수 있는 지를 하위 수준의 함수 단위로 기능을 분해한다. |
| 기능 검토 | 분석된 기능이 검토된 요구 사항을 완전히 지원하는 지 검토한다. |

4. 요구 사항 표현 방법

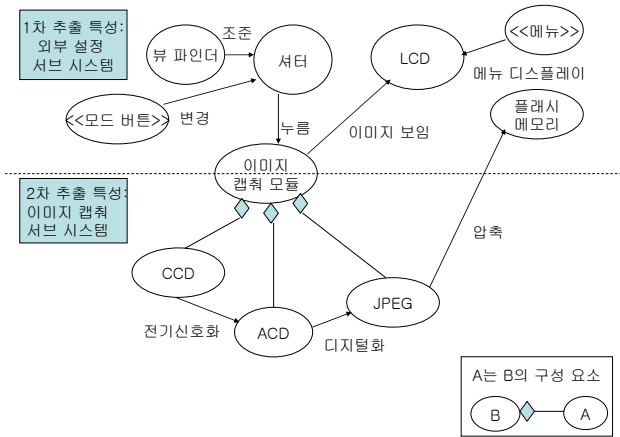
4.1. 의존성 그래프(상위 요구 사항 표현 방법)

상위 요구사항은 텍스트 형태로 표현한 후 유스 케이스 시나리오로 받아 들여 제품의 기능에 관여하는 객체의 이름과 행위의 이름을 단어 단위로 분석한다. 식별된 객체와 행위는 객체와 행위 간의 의존성을 실행 흐름으로 표현하기 위해서 그래프로 표현하고 의존성 그래프라고 명명한다. 의존성 그래프는 노드와 레이블에 붙어 있는 에지로 표현한다.

노드(node)는 특성 객체를 표현하고 에지(edge)에 붙어 있는 레이블은 특성 행위를 표현한다. 에지는 특성 행위의 순서를 지시할 수 있기 때문에 그래프는 요구 사항의 구성 객체와 행위의 이름을 연결 관계(이하 의존성 그래프)로 표현하고 요구 사항 전체를 구조화할 수 있도록 본 논문에서 제안한다.

그림 1 은 디지털 카메라에 대한 의존성 그래프를 나타낸다. 가운데의 점선을 경계로 디지털 카메라에 대해 1 차적으로 추출된 객체와 객체 간에 이뤄지는 행위를 나타낸다. 예지는 객체 간에 행해지는 행위와 행위의 순서를 나타내며 1 차적으로 추출된 객체의 세부 구성 요소를 찾고 2 차로 추출된 객체인 'JPEG'의 구성 요소와 관련 있는 객체 '플래시 메모리'를 찾아 표현한다.

의존성 그래프는 요구 사항으로부터 시스템의 기능을 수행하는데 필요한 필수 객체를 분석하는데 이용된다. 생성된 그래프를 반복적으로 정제하여 식별된 객체들은 다시 분해하고 그룹핑 하는데 의미적으로 분해할 수 없는 수준의 객체는 특성 객체로 정의된다. 그리고 정제 과정을 거치면 특성 객체와 특성 객체에 대한 가변성이 분석된다. 공통적으로 사용되는 객체에 끼워져 결합될 수 있는 플러그인 형태로 다른 플러그인의 다양성이 결합해 결정되는 객체는 가변점이라고 정의하는데 의존성 그래프를 표현해 나가면서 변경 가능한 부분은 가변성으로 정의된다. 그림 1 의 '메뉴'나 '모드 버튼'이 하나의 예가 된다. '이미지 캡취 모듈'은 CCD, ACD, JPEG 과 같이 이미지를 캡취하는 객체들을 구성 요소로 가짐을 표현한다.



(그림 1) 의존성 그래프

반복적인 정제로의 객체와 행위는 정돈된 형태로 정리되고 관리될 필요가 있다. 따라서 우리는 테이블 형태로 의존성 그래프를 유스 케이스-가변성 정의의 그래프로 표현한다.

4.2. 유스 케이스-가변성 그래프(상세 요구 사항 표현 방법)

유스 케이스-가변성 정의의 그래프는 특성 객체 리스트, 특성 객체의 범주, 기본 설정 값, 참조 설정 값, 관련 유스 케이스 이름, 지원하는 가변점 리스트, 가변점 타입의 추가 정보로 구성된다. 표현 방법이 바뀐 유스 케이스-가변성 정의의 그래프는 시각적으로 처리되었을 때 그래프 형태나 텍스트 형태로 보일 수 있다. 유스 케이스-가변성 형태의 표현은 다양한 시각적인 정보로 표현될 데이터들을 통합적으로 관리할

수 있고 사용자의 추가적인 분석 정보를 포함하는데 확장성을 갖게 된다.

유스 케이스-가변성 정의의 그래프는 정제를 거쳐 요구 사항 분석 정보를 검토하기 위한 테스트 정보를 유도할 수 있다. 특정 객체(이하 특성 객체)에서 시작하여 연결된 타 객체(이하 특성 객체 리스트)간의 행위(이하 특성 행위) 정보들을 하나의 경로로 식별하면서 테스트에 필요한 경로로 이용할 수 있다. 테스트 경로는 경우에 따라 최적의 경우와 최악의 경우를 찾는데 도움이 될 수 있다. 특히 최악의 경우는 요구 분석 단계에서 분석과 정제의 과정을 거치며 기술적으로나 개발 환경적으로 필요한 리소스를 많이 필요로 하는 객체와 경로에 대해 이뤄진다. 이는 충분히 식별된 요구 분석 정보에 기반하므로 요구 분석과 연관된 설계, 구현, 시험, 통합, 배포의 개발과정의 수행과 연관되게 된다.

그림 2 는 유스 케이스-가변성 정의의 그래프를 표현한다. 각 색인은 특성 객체(FO), 특성 행위(FB), 특성 객체 리스트(LFO), 관련 있는 특성 객체(FC), 특성 범주, 특성 객체의 기본 설정 값(DV), 참조 설정 값(RV), 유스 케이스 명(UC), 가변점(VP), 가변점의 타입(T)으로 표현되어 보여진다.

| 번호 | FO | FB | LFO | FC | DV | RV | UC | VP | T |
|-----|--------------|--------|-----------|----|-------|-------|-----|-------------|---|
| 0 | 전원 | 켜기 | &1 &3 ... | A | OFF | | | | |
| 1 | 모드 | 변경 | &2 &3 ... | D | 활영 | | UC1 | VP3 VP4 ... | D |
| 2 | 셔터 | 누름 | | O | | | UC1 | | |
| 3 | 메뉴 | 조작 | &2 &1 | D | | | UC1 | VP1 VP2 ... | O |
| 4 | CCD | 이미지 원기 | &5 | D | | | UC2 | | |
| 5 | ACD | 디지털 화 | &6 | D | | | UC2 | | |
| 6 | JPEG | 압축 | &7 | D | | | UC2 | | |
| 7 | 메모리 | 저장 | | D | empty | 남은 용량 | UC2 | | |
| 8 | 시리얼 버스 연결 소켓 | 업로드 | | D | | | UC3 | | |
| 9 | PC | | | | | | UC3 | | |
| VP1 | 한글 | 출력 | | D | | | UC4 | | |
| VP2 | 영어 | 출력 | | A | | | UC4 | | |
| VP3 | 활영 모드 | 설정 | | D | | | | | |
| VP4 | 디스플레이 모드 | 설정 | | D | | | | | |

(그림 2) 유스 케이스-가변성 정의의 그래프

| UC | | VP | | | | |
|-----|----------|-----|-----|-----|-----|----|
| UC1 | 외부설정시스템 | VP1 | VP2 | VP3 | VP4 | .. |
| UC2 | 캡취 시스템 | | | | | |
| UC3 | 업 로딩 시스템 | | | | | |
| UC4 | 메뉴 조작 | | | | | |

(그림 3) 유스 케이스 별 가변점 분석

그림 3 은 분석된 유스케이스 별로 존재하는 가변점을 구할 때 임시적으로 사용되고 분석된 가변점 정보는 유스 케이스-가변성 정의의 그래프를 통해 표현될 수 있다.

요구 사항 분석은 제품군에 대한 요구 분석일 경우 제품군이 속해 있는 도메인에 대한 분석을 의미한다. 전체적으로 요구 사항 분석은 시스템에 대한 아키텍처를 설계하고 구현하며 아키텍처의 구성 요소를 설계하여 구현할 수 있도록 하는 기본 정보가 된다. 따

라서 분석 정보에 대한 확인 작업이 분석 이후 작업에 대한 시작으로 이뤄질 수 있도록 해야 한다.

유스 케이스-가변성 정의 그래프를 통한 테스트는 특성 객체를 대상으로 시작하여 가변성을 선택하여 이뤄질 수 있다. ‘전원’ 객체에서 시작하여 ‘켜기’ 행위가 이뤄지는 유스 케이스에 대해 다음으로 동작될 행위는 ‘모드 변경’이나 ‘셔터 누름’과 관련되고 각각의 행위는 ‘특성 객체 리스트(LFO)’에 표시되어 있는 순서를 따라 판별될 수 있다. 이런 식으로 관련 객체와 객체의 행위를 식별하는 과정은 최종적으로 끝나쳐 지는 객체를 결정하여 그 객체까지 행위 리스트를 검토함으로써 연결 관계를 식별하고 이에 기반한 다양한 시험 계획을 작성할 수 있다.

이와 같이 이뤄지는 테스트 경로 식별 및 검증 작업에 분석된 가변성을 테스트 경로에 각각 대입하듯이 수행하면 특정 가변성이 주어졌을 때 수행될 행위가 어떤 절차로 수행될 지 판별할 수 있다.

유스 케이스-가변성 정의 그래프의 정보가 풍부해질수록 생산하고자 하는 제품의 기능과 기능에 대한 테스트 계획을 보다 명확히 정의해 나갈 수 있다.

5. 평가

일련의 제품군은 공통적으로 유사한 기능에 다양하게 차별화되는 기능이 결합되어 이뤄진다. 따라서 제품군의 각 시스템은 개별적으로 아래와 같이 도메인 분석, 시스템 아키텍처 설계, 컴포넌트 설계, 구현, 검증과 확인, 배포의 개발 단계를 거치게 된다.



(그림 4) 개발 단계

각 단계가 모두 중요하지만 제품에 대한 정확한 분석 및 영역 확정이 어떤 단계로든 나아가 단계에 대한 기반 정보가 되기 때문에 기반 정보의 관리 방법은 중요하다. 각 단계는 순차적으로 이뤄지지만 실제적으로는 검증과 확인 과정이 각기 다른 단계의 중간과 종료 시점에 포함되어 이뤄지게 된다. 따라서 검토와 확인 과정은 각 단계의 작업 수준에 맞춰 신속하고 정확하게 수행 기준을 가지고 수행될 필요가 있다.

특히 요구 분석은 도메인 분석 단계에서 요구 사항에 대한 분석을 통해 테스트 요구 사항을 도출하고 테스트 경로를 식별하여 테스트 계획서를 작성할 수 있는 초기 단계가 된다. 요구 분석 정보에 기반한 테스트 계획서는 테스트 경로 식별 작업과 밀접한 연관성을 가지고 개발 각 단계에 필요한 테스트 계획과 테스트 케이스 생성에 대한 일관성 있는 기반 정보가 되기 때문에 테스트 기반의 개발을 지원할 수 있다.

6. 결론

본 논문은 우선순위 그래프를 이용한 제품군에 대한 기능을 분석하여 표현하고 테스트 경로를 찾는 방

법에 대해 제시했다. 제품의 기능을 특성으로 표현하고 각 특성을 객체와 행위로 식별하여 표현된 그래프는 유사하지만 다양한 기능을 갖는 제품에 대한 표현이 된다. 제품의 기능에 대한 표현은 그래프로 표현될 수 있고 기능에 대한 표현 그래프는 기능에 대한 유사성과 다양성을 표현한다. 그래프에 설정된 값과 범위는 분석 단계에서 식별된 유스케이스를 테스트할 수 있는 기본 정보가 된다.

개발 각 단계의 기반이 되는 도메인 요구 분석은 특정 원칙과 전략을 필요로 하나 어떤 대상을 요구 분석의 대상으로 하는 지 어느 정도의 정보를 포함해야 하는 지에 관해서는 본 연구에 포함하지 않았다. 이는 매우 주관적이라 도메인 별로 분석 대상으로 고려할 수 있는 대상과 범위, 초기 설정 값에 차이가 존재하고 그 차이가 크기 때문이다.

초기에 비즈니스 수행 경로로 시작된 경로는 기능을 테스트 하기 위한 일련의 절차로 추출되는 테스트 경로의 에지 수에 따라 그리고 경로 상에 존재하는 객체에서 필요로 하는 리소스의 양에 따라 최악의 깊이를 갖는 테스트 케이스와 최적의 깊이를 갖는 테스트 케이스를 예측해 볼 수 있었다. 이를 통해 테스트 케이스 도출과 관련된 연구의 가능성을 알아 볼 수 있었으나 보다 깊이 연구될 필요가 충분히 존재한다.

수행 경로의 식별과 깊이의 비교는 수행 환경과 수행 리소스(인력, 시간, 개발 자원)에 대한 분배 전략을 세울 수 있으므로 정해진 분석 시간 동안 충분히 요구 사항을 분석하고 상세화하는 작업이 필요하다.

본 논문에서 제시한 방법은 텍스트 형태의 요구 사항 기술에서 시작하여 시스템 개발에 참여하는 시스템 분석가, 시스템 개발자, 테스트 설계자, 프로젝트 관리자에 이르는 모든 이해 당사자들간의 분석 정보 공유와 통합 검토를 위한 시스템에서 필요한 변형을 통해 다양하게 사용될 수 있을 것으로 예측한다.

참고문헌

- [1] Giancarlo Succi, Jason Yip, Witold Pedrycz, "Holmes: An Intelligent System to Support Software Product Line Development", ICSE 2001, May 2001, pp. 829-830.
- [2] Together,
- [3] A. Fantechi, S. Gnesi, I. John, G. Lami, J. Dorr., "Elicitation of Use Cases for Product Lines", Int. Workshop on Product Family Engineering, Nov. 2003, pp.152-167.
- [4] Lang Theodore James, "Method for Analyzing Product Information", 2004.
- [5] Jack Ferguson, "Crouching Dragon, Hidden Software: Software in DoD Weapon Systems", IEEE Software, July 2001, pp. 105-107.