

소프트웨어 개발 프로세스를 대상으로 수행하는 추적성 분석의 세부 관계 정의

김재엽, 이동아, 유준범

건국대학교 컴퓨터공학
서울시 광진구 능동로 120 건국대학교
radic2510@gmail.com, {ldalove, jbyoo}@konkuk.ac.kr

요약: 본 연구에서는 표준 규격을 바탕으로 소프트웨어 개발 프로세스 전반에 걸쳐 수행되는 추적성 분석에서 사용되는 세부 관계를 정의한다. 기존의 추적성 분석에서 항목들 간의 연결로 추적을 표현하지만 다양한 관계의 종류를 분류할 필요가 있기 때문에 세분 관계를 정의하였다. 정의한 세부 관계를 ARINC 653 과 Qplus-AIR 요구사항 명세의 추적성 분석에 적용하여 효용성을 보여준다.

핵심어: 추적성 분석, ARINC 653, Qplus-AIR

1. 서론

소프트웨어 개발과정에서 검증과 관리, 유지보수의 중요성이 점차 증가함에 따라 추적성 분석에 대한 중요도가 증가하고 있다. 추적성 분석은 소프트웨어 개발과정에서 생산되는 산출물들이 어떠한 요구사항 요구부터 생산되는 것인지 추적관계를 통해 분석하는 것을 말한다[1].

기존의 추적성 분석에서는 특정 산출물이 어떠한 요구사항으로부터 생산된 것인지 일괄적인 표현방식을 사용하여 표현하는 것이 일반적이다. 하지만 실제 표준 규격을 바탕으로 수행되는 소프트웨어 개발 프로세스에서의 추적성 분석에서는 표준과 요구사항, 요구사항과 설계 간의 다양한 관계가 존재하기 때문에 일괄적으로 표현하는 것으로는 한계가 있다. 따라서 추적성 분석에서 다양한 관계를 표현하기 위한 세부 관계를 정의할 필요가 있다.

본 논문에서는 표준을 바탕으로 수행되는 소프트웨어 개발 프로세스의 추적성 분석에서 사용되는 세부 관계에 대해 정의한다. 그리고 정의한 관계의 효용성을 확인하기 위하여 ARINC 653 과 Qplus-AIR 간의 추적성 분석에 적용하고 그 결과를 확인하여보았다.

2. 추적성 분석

추적성 분석은 시스템 요소를 위한 요구사항과 계층관계에 있는 요구사항이나 설계, 구현 등과 같은 다양한 산출물과 연결하는 프로세스이다[2]. 추적성 분석은 개발 프로세스 전반에 걸쳐 수행되며 좁은 범위로는 문서 간의 분석을 수행해서 요구사항이 모두 반영되었는지 확인하는 용도로 사용되지만 넓은 범위로는 구현된 시스템이 초기 요구사항으로부터 어떤 설계와 구현과정을 거쳐 생성된 것인지에 대한 추적이 가능하다. 복잡한 시스템을 대상으로 하는 추적성 분석 결과는 정형 검증을 위한 속성을 도출하는 용도로 사용하는 것이 가능하다.

3. 추적성 분석 세부 관계

추적성 분석에서 사용되는 세부 관계를 5 가지로 나눠서 정의하여 <표 1>에 정리하였다. Reference 와 Application 은 기존의 추적성 분석에서 연결되는 관계를 세분화 한 것이고 No Relationship 의 세 가지 관계는 기존의 추적성 분석에서 추가로 고려되는 관계이다.

Reference 와 Application 은 기존의 추적 분석에서 연결되는 것을 세분화 한 관계이다. Reference 의 경우 요구사항 명세에서 표준을 직접적으로 언급하고 참조하라고 되어 있는 경우가 해당된다. Application 은 표준에서 요구하는 부분을 프로젝트에 적합하게 바꾼 경우에 해당된다. 예를 들어 임베디드 시스템의 경우 시간과 관련된 부분에서는 보드의 특성을 고려해야 하기 때문에 추가적인 고려가 필요하게 되는데 이러한 경우가 Application 에 해당한다. Reference 는 표준을 그대로 사용하는 관계이지만 Application 은 표준을 바꿔서 사용한다고 할 수 있기 때문에 요구사항에 대한 추가적인 검증이 요구될 수 있는 부분이다. 관계를 세분화 함으로서 집중적으로 검증을 수행하는 부분을 선택하기 용이해진다.

표 1 추적성 분석 세부 관계

이름	설명
Reference	표준과 같거나 표준을 참조를 표현한 경우의 관계
Application	표준의 항목을 프로젝트에 적합하게 변경하여 사용한 경우의 관계
No Relationship (Selection)	다양한 방법 중 일부를 선택하여 개발이 가능한 경우에서 다른 방법을 선택한 경우
No Relationship (Creation)	개발 프로세스에서 새롭게 생성된 경우로 요구사항으로부터 생성된 항목이 아닌 경우
No Relationship (Omission)	표준의 항목이 추적성 분석에서 연결되는 항목이 없는 경우 중 필히 연결되어야 하는 항목인 경우

No Relationship 은 추적성 분석으로 추적이 수행되지 않는 부분이다. 이에 해당하는 관계는 3 가지로 구분될 수 있는데 표준에서 다양한 방법 중 일부를 선택하여 개발이 가능한 경우에서 선택되지 않은 항목(Selection)과 산출물에서 새롭게 생성된 항목(Creation) 그리고 요구사항을 만족하지 못한 경우(Omission)이다. Selection 의 경우 표준에서 언급하긴 하지만 선택적으로 요구되는 부분이다. 예로 파티션 간 통신이 다양한 방법으로 구현될 수 있고 표준에서도 다양한 방법이 항목으로 기술되어 있지만 실제 구현에서는 특정 방법을 사용해서 구현하는 것이 이에 해당한다. Creation 의 경우 표준에는 존재하지 않지만 요구사항 명세에서 새롭게 생성되는 항목으로 대표적인 예로는 구현된 시스템의 하드웨어와 관련된 부분이 속할 수 있다. 두 관계는 기존의 추적성 분석에서는 나타나기 어려운 부분으로 추적성 분석을 수행한 사람이 아니라면 해당 부분이 Omission 또는 추적성 분석을 수행하지 않은 부분으로 착각하여 추가적인 비용을 낭비하게 된다. 추적을 나타낼 수 없는 부분에서도 이러한 관계를 표현하여 비용을 절약할 수 있다.

4. 적용 및 분석

본 연구에서 정의한 추적성 분석 세부 관계가 실제 시스템 개발 프로세스에서 다양한 연결 결과를 보여주는지 확인하기 위해서 ARINC 653 표준 규격 [3]과 이를 바탕으로 만들어진 Qplus-AIR [4]의 요구사항 명세서 간의 추적성 분석 결과를 분류해 보았다. ARINC 653 은 195 개 Qplus-AIR 의 요구사항 명세서는 179 개의 항목을 가지는 문서이다.

추적성 분석의 대상이 되는 항목만을 추려내기 위한 기존의 추적성 분석을 수행하였으며 ARINC 653 의 항목을 195 개에서 155 개로 Qplus-AIR 의 항목을 179 개에서 169 개의 항목으로 추려내었다. <표 2>는

추적성 분석 결과를 위에서 정의한 5 가지 세부 관계로 분류한 결과이다. 총 270 개의 관계가 생성되었으며 이 중 65 개의 항목은 논문에서 정의한 관계에 의해서 추적성 분석이 수행된 것이다. 이 중 Omission 으로 관계를 분류해 놓은 27 개의 관계에 대해서는 추가적인 확인이 필요한 부분이다.

표 2 세부 관계 적용 결과

관계	개수
Reference	37
Application	168
No Relationship (Selection)	5
No Relationship (Creation)	33
No Relationship (Omission)	27

총 270 개의 관계 중 No Relationship 에 해당하는 65 개의 항목은 기존의 추적성 분석에서는 모두 Omission 인 경우로 해석되어 추가적인 비용을 요구하게 되는 부분이다. 세부 관계의 경우 이러한 경우를 포함하여 추적성 분석을 수행하기 때문에 비용의 절약이 가능하다. 하지만 현재 본 논문에서 정의하는 Omission 의 경우는 추적성 분석을 수행할 당시 기준의 설정이 적절하기 못한 경우 생길 수 있다는 점과 요구사항에서 의도적으로 생략한 부분에 대해서는 확인 할 수 없다는 점 등의 한계가 존재하기 때문에 추가적인 세분화가 요구된다. Reference 와 Application 은 기존의 추적성 분석에서도 연결이 가능하지만 세부 관계를 통해서 개발 프로세스 내의 다른 단계에서 활용할 수 있었다. Qplus-AIR 을 대상으로 정형검증을 수행하기 위한 검증 속성을 추적성 분석 결과를 활용하여 생성하였다. Reference 관계 보다 Application 관계를 검증하는 것이 더 중요하기 때문에 이를 위한 검증 속성 도출에 더 많은 비용을 소비하였다.

5. 결론 및 향후 연구

본 연구에서 정의한 5 가지 세부 관계는 추적성 분석에서 요구사항과 산출물 간의 관계 표현을 세분화하여 분류할 수 있다는 점에서 의미가 있다. 특히 표준을 기반으로 수행한 개발 프로세스에서 요구사항을 만족 여부에 대한 분석을 기존의 추적성 분석 보다 세분화하여 표현하는 것이 가능하기 때문에 검증 부분에서 얻을 수 있는 이점이 있다.

본 연구는 ARINC 653 표준 규격과 Qplus-AIR 의 요구사항 명세서 간의 추적성 분석 수행을 바탕으로 세부 관계를 정의하였지만, 향후 연구에서는 세부 관계 정의를 보다 세분화하여 정의하고 이를 기반으로 추적성 모델까지 정의하고자 한다. 또한 설계 및 구현까지의 추적성 분석을 수행하고 분석 결과를 평가하고자 한다. 정형검증을 위한 속성 추출 과정에서

세부 항목을 적용한 추적성 분석 결과의 효용성 분석을 수행하고자 한다.

사 사

본 논문은 한국전자통신연구원의 “오류없는 시스템 통합을 위한 안전우선분산 모듈형 SW 플랫폼” 사업의 지원으로 연구한 결과입니다.

참고문헌

- [1] 이준기, 조혜경, 고인영, “요구사항 온톨로지 기반의 시맨틱 태깅을 활용한 산출물의 재사용성 지원을 위한 요구사항추적 방법,” 정보과학회논문지 : 소프트웨어 및 응용, 35(6), 357-365, 2008
- [2] Antoniol, Giuliano and Canfora, Gerardo and Casazza, Gerardo and De Lucia, Andrea and Merlo, Ettore, “Recovering traceability links between code and documentation,” Software Engineering, IEEE Transactions on, Vol.28, No.10, pp.970-983, 2002
- [3] Avionics Application Software Standard Interface, ARINC Specification 653P1-3
- [4] 최두열, 장순용, 김재용, 류춘하, 조인제, 김태호, “ARINC 653 을 지원하는 Qplus-AIR 를 활용한 무인기용 비행제어 소프트웨어 개발 연구,” 한국항공우주학회 2012 춘계 학술대회, pp.1176-1180, 2012