Safety Case 패턴의 분류 카테고리에 따른 인스턴스 생성

정세진 ○, 김의섭, 유준범

건국대학교 컴퓨터공학과

{jsjj0728, atang34, jbyoo}@konkuk.ac.kr

An Instance Generation of the Safety Case Pattern by the Classification Category

Sejin Jung ○, Eui-Sub Kim, Junbeom Yoo

Division of computer science and engineering, Konkuk university

요 약

Safety case는 시스템이 용인되는 수준의 안전성을 갖췄는지 보이기 위한 명시적이고 구조적인 논증 구조를 표현하는 기법이다. Safety case를 통한 논증 구조 작성에는 많은 노력이 필요해 반복적으로 재사용 되는 구조/내용 등을 safety case pattern을 통해 재사용하는 경우들이 있다. 이 때 safety case pattern을 활용해 인스턴스(instance)를 생성할 때 제공되는 패턴에 따라 인스턴스 생성에 여러 차이점들이 있다. 본 논문에서는 이전에 수행한 패턴 작성 범위 분류에 따라 safety case pattern의 인스턴스를 생성하는 방법 및 그 관계에 대해 제안한다. 또한 사례 연구를 통해 본 논문에서 제안하는 인스턴스 생성 방법의 결과를 확인하였다.

1. 서 론

안전필수 시스템(Safety-critical system)은 시스템의 안전성을 증명하기 위해 표준, 법규, 가이드라인에 준수해 다양한 안전 활동(safety activity)를 수행하며 개발된다. Safety case는 다양한 안전 활동의 결과물 만으로는 시스템의 안전성을 판단하기 어렵기 때문에 명시적이고, 구체적인 논증 구조를 통해 시스템이 용인되는 수준의 안전성을 갖췄는지 확인하는 기법이다. Safety case pattern은 반복되는 구조를 재사용하기 위해 제안된 방법으로 기본 논증 구조(structure) 외에 여러 패턴 language로 구성된다.

이전 연구에서는 기존에 제안된 safety case pattern을 기반으로 하여 패턴의 작성 수준, 범위에 따라 4 종류의 분류 기준을 제안하였다[1]. Safety case pattern을 활용하여 그 인스턴스(instance)를 생성하기 위해서는 실제 대상이 되는 시스템/소프트웨어에 대한 내용을 작성하는 instantiation이 필요하다. 이 때 각 패턴의 분류에 따라 필요한 인스턴스화 수준 및 정의와 작성 범위가 각각 다르게 나타난다. 본 논문에서는 이에 따라 4 종류로 구분되는 각 분류에 따라 safety case instance 생성의 관계와 그 방법의 정도에 대해 제안한다. 또한 각 분류 레벨에 따라 인스턴스 생성 시 패턴과 간섭되는 정도에 대해 논한다.

2. Safety case pattern

Safety case pattern은 시스템의 안전 증명을 위한 safety case 구조에서 자주 사용되는 공통된 내용, 구조를 재사용하기 위한 패턴화된 접근법이다. 따라서 정형화된 구조 제공이 중요하며 이를 위해 [2] 에서는 pattern language를 제안해 주요 내용을 명확하게 제공하도록 하도록 하였다. [2]에서는 GSN을 사용하여 structure 항목의 작성을 제안하고 있으며 추가적으로 다중 요소(multiplicity), 반복 구조(feedback loop)와 같이 패턴에 필요한 요소들을 포함하고 있다. 다음 <그림 1>은 safety case pattern에 대한 예제이다. 아래 그림과 같이 패턴은 기본 요소 및 instantiation을 통해 작성될 항목 요소 (e.g. '*{HSFM}*')들을 가지고 구성된다.

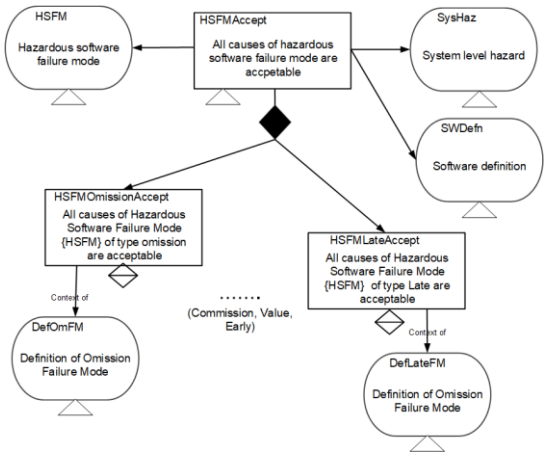

그림 1. Safety case pattern structure의 예제 [3]

3. Safety case pattern의 인스턴스 생성

[1]에서 제안하는 safety case pattern들의 분류 카테고리는 다음 <표 1>과 같다. 패턴에서 사용되는 작성 범위에 따라 4 단계로 분류하고 있으며, 각 분류에 따라 제공되는 내용 및 구조, 내부 컨텐츠의 세부 구체화 정도가 다르게 구성된다.

3.1 단계 분류 별 인스턴스 생성의 관계

Safety case pattern을 통한 인스턴스 생성은 2 단계로

진행될 수 있다. 첫 번째는 대상 시스템/소프트웨어에 대한 argument가 되도록 패턴의 내용을 채우는 부분으로 인스턴스를 생성한다고 할 수 있다. 특히 safety case pattern 에서 {SW::target SW}와 같이 인스턴스 생성 대상을 지칭하는 부분들을 작성한다. 다음으로는 대상의 실제 안전 활동 산출물을 활용해 논증 구조를 완성하는 단계이다. 이 부분에서 추가적인 strategy와 solution notation 들이 사용되어 safety case argument를 완성하게 된다. 본 논문에서는 이를 각각 instance 작성과 argument 완성 단계로 정의하였다.

표 1. Safety case pattern의 분류 카테고리[1]

No.	Classification	Description
1	Structural composition	Notation들의 structure만으로 구성
2	High-level contents composition	높은 수준에서 abstraction된 내용만 제공
3	Concrete contents composition	대상 domain에 specific한 내용을 포함한 abstraction
4	Detailed contents composition	가장 낮은 수준의 abstraction으로 safety case의 인스턴스에 근접한 패턴

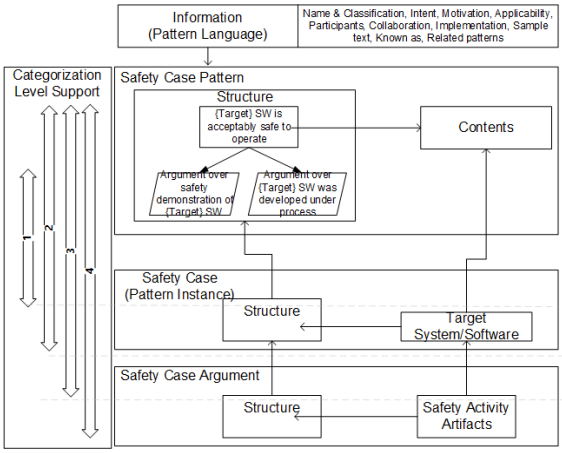

그림 2. Safety case pattern 과 instance 생성 관계

<그림 2>는 각 카테고리에 따라 인스턴스 생성의 관계에 대한 그림이다. <표 1>에서 4번으로 분류된 패턴의 경우에는 가장 하위 레벨의 상세한 요구사항 내용까지 모두 패턴에 포함되어 제공되기 때문에 모든 범위를 커버할 수 있다. 반면 1 번으로 분류된 경우에는 notation의 구조만을 제공하거나 기계적으로 자동 생성하기 위한 패턴들이 포함되기 때문에 structure와 타겟 위주로 커버되어 인스턴스를 생성하는 것이 특징이다. 2, 3번으로 분류된 패턴은 각각 instance 단계와 argument 단계의 일부에 포함된다. 이처럼 각 safety case pattern은 패턴의 제공되는 내용/구조에 따라 인스턴스 생성에 서로 다른 관계를 가지고 있음을 확인할 수 있다. 따라서 패턴을 이용해 최종 단계의 인스턴스를 생성하기 위해서는 이에 맞는 추가적인 내용의 작성이 필요하며 작성되어야 할 내용의 범위/수준은 패턴의 카테고리에 따라 차이가 있다. 다음 장 에서는 이에 대해 체계적인 작성을 위해 패턴의 분류 카테고리 별 인스턴스 생성 방법에 대해 소개한다.

3.2 레벨 분류 별 instance 생성 방법

위 절에서 밝힌 것과 같이 각 분류 단계에 속하는 패턴들은 커버하는 범위가 모두 다르기 때문에 패턴을 이용해 인스턴스를 생성하기 위해서는 적절한 instantiation을 통해 수행되어야 하며, 이는 각 패턴의 종류, 범위에 따라 모두 달라진다. 본 논문에서는 safety case의 인스턴스 생성 시 체계적인 작성 및 자동화 프로세스를 위해 [1]에서 분류한 레벨에 따라 각 방법에 대해 제안한다. <그림 3>은 본 논문에서 제안하는 인스턴스 생성 과정에 대한 그림으로 각 카테고리 별 서로 다른 과정을 가지고 있다.

우선 공통적으로 작성 대상이 되는 타겟과, 사용할 패턴을 선택하는 과정이 진행되며, 카테고리 1의 패턴의 경우는 structure만으로 구성되거나 기계적인 생성에 대한 패턴이기 때문에 보조적인 역할로 활용이 가능하며 패턴의 주된 내용은 직접 작성해야 한다. 따라서 작성법, 구성법을 활용해 직접 safety case를 개발하는 과정만으로 진행된다.

그 외의 카테고리 들은 Step 3, 4를 거쳐 단계적으로 패턴의 instantiation 요소 및 multiplicity, loop notation과 같은 element들을 대상에 맞게 수정하는 작업을 진행한다. 이 때 해당 요소 들은 여러 번 나타날 수 있기 때문에 반복해 진행한다. Step 3, 4를 거쳐 <그림 2>에 나타난 safety case pattern instance 작성이 진행된다.

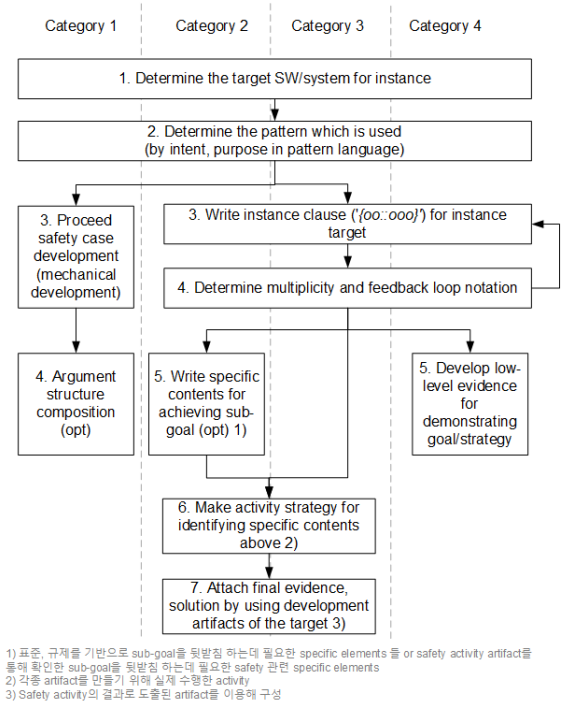

그림 3. Safety case pattern의 카테고리별 인스턴스 생성 과정

공통적으로 진행되는 과정 이후에 카테고리 2에 해당하는 패턴은 Step 5를 통해 abstraction sub-goal을 뒷받침하기 위한 specific contents를 작성해야 한다. Specific contents는 패턴의 하위에 나타난 sub-goal의 내용을 분해하여 부 목표

(Sub-goal)의 충족 여부를 쉽게 확인할 수 있도록 뒷받침하기 위한 내용으로써 표준이나 개발 산출물을 통해 작성한다. 카테고리 2에 속하는 패턴들은 높은 수준에서 추상화된 내용으로 구성되기 때문에 실제 확인할 구체적인 요소들을 작성하는 것이 필요하다.

다음으로는 전략을 작성하는 부분으로 위에서 작성한 specific contents를 어떤 방법으로 확인하여 달성되었는지를 뒷받침하는 실제 activity가 포함된 전략을 작성하는 과정으로 카테고리 2, 3 모두 같은 과정으로 진행된다. 이 때 실제 수행한 액티비티들을 위주로 구성하여 논증 구조의 최종 결과로 실제 도출된 산출물을 기반으로 evidence 가 구성될 수 있도록 해야 한다.

카테고리 4의 경우는 액티비티를 포함한 요구사항 등 많은 내용이 패턴을 통해 제공되기 때문에 직접적으로 인스턴스 생성과정을 거쳐 argument 수준의 논증구조 생성이 가능하다. 다음 <표 2>는 인스턴스 생성의 주요 과정의 작성 정보에 대한 표이다.

표 2. 주요 과정의 작성 정보

Category	Process	Description	From
2	5	Specific contents/element for supporting goal achievement	Std. or Safety artifact
2	6	Activity to achieve sub-goal	Safety artifact
2	7	Specific evidence	
3	6	Activity to achieve sub-goal	
3	7	Specific evidence	
4	5	Specific evidence	

4. 사례 연구

Safety case pattern을 활용해 인스턴스를 사용하는 간단한 예제는 다음과 같다. 본 논문에서는 안전필수 시스템 중 하나인 원자로 보호 시스템의 BP 소프트웨어를 대상으로 패턴을 활용한 인스턴스 생성을 수행하였다. <그림 4>는 안전 관련 소프트웨어의 안전 논증을 위한 safety case pattern으로 안전 증명을 위해 기능적 오류가 없어야 함을 목표로 하고 있는 패턴의 일부이다. 해당 패턴은 <표 1>의 카테고리에서 2 수준에 속하도록 개발되었다.

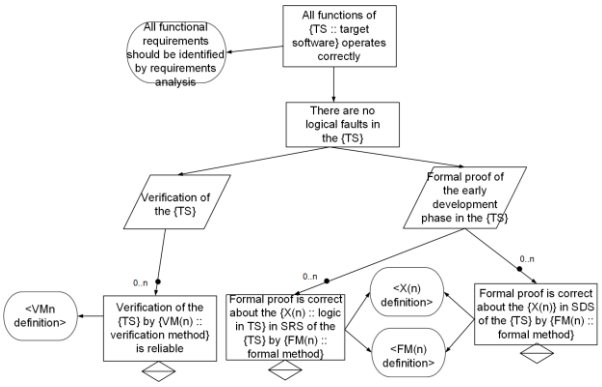

그림 4. Safety SW를 위한 safety case pattern 예제

<그림 5>는 패턴을 이용해 BP SW의 인스턴스를 생성한 결과에 대한 그림으로 본 논문에서 제안하는 프로세스에 따라 수행된 결과이다. 인스턴스 생성은 [4]에 나타난 최종 솔루션을 바탕으로 진행하였다. 특히 step 5를 거치며 specific contents로는 테스팅의 종류를 추가해 검증이 진행되었음을 뒷받침하도록 하였고, step 6의 액티비티로는 보고서 확인으로 작성되었다. 이는 실제 대상과 안전 활동 결과물에 따라 변화되는 부분이다.

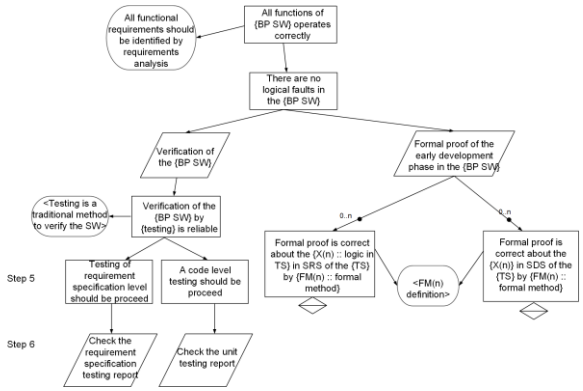

그림 5. Safety case pattern의 인스턴스 생성 예제

5. 결론 및 향후 연구

이처럼 본 논문에서는 safety case pattern의 분류 카테고리별 인스턴스 생성의 관계에 대해 확인하고 그 프로세스를 제안하였다. 기 연구된 분류 카테고리 별로 각 패턴을 인스턴스 생성에 이용하기 위해서는 여러 활동이 추가되어야 함을 확인할 수 있었고, 본 논문에서 제안하는 방법을 통해 인스턴스 생성을 진행할 수 있음을 확인하였다. 향후 효과적인 safety case pattern 개발을 위해 각 카테고리 별로 패턴을 생성하고, 구성하는 방법 및 제약사항에 대해 연구할 계획이다.

Acknowledgement

이 논문은 2019년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업의 결과입니다. (NRF-2017R1D1A1B03030065)

참고 문헌

[1] 정세진, 손준익, 유준범, "원자력 발전소 안전 소프트웨어의 safety case pattern 작성을 위한 문헌 리뷰 기반의 pattern 작성 범위 분류", 한국정보과학회 2018 한국컴퓨터종합학술대회, pp.577-579, 제주도 국제컨벤션센터, 6.20-6.22, 2018.

[2] T. P. Kelly, J. A. McDermid, “Safety Case Construction and Reuse using Patterns,” Safe Comp 97, pp.55-69, 1997.

[3] R.A. Weaver, “The Safety of Software - Constructing and Assuring Arguments,” Ph. D. Thesis, University of York, 2003.

[4] 권기춘, 이장수, 박기용, 지은경, “원전 디지털 원자로보호계통 소프트웨어 안전보증 패러다임” 한국정보과학회 2016 한국컴퓨터종합학술대회, pp.642-644, 2016.