TC2SMV: UML State Chart Diagram에서 생성된 테스트 케이스를 SMV 입력 프로그램으로 변환하는 CASE 도구

윤상현 ○, 조재연, 유준범
건국대학교 정보통신 대학
pctkdgus@konkuk.ac.kr, tm77@konkuk.ac.kr, jbyoo@konkuk.ac.kr

TC2SMV: A CASE tool for Translating SMV Input Program from Test Cases Generated from State Chart Diagram in UML

Sanghyun Yoon○, Jaeyeon Jo, Junbeom Yoo
College of Information and Communication, Konkuk University

요 약

소프트웨어 시스템의 안전성을 보장하기 위해 여러 안전성 분석 기법들이 사용되고 있으며, 안전성 분석의 결과인 안전성 요구사항을 시스템이 만족하고 있는지 확인하는지 확인하는 안전성 평가과정이 수행되고 있다. 안전성 평가는 많은 시간과 비용이 드는 문제점을 가지고 있다. 본 논문은 안전성 평가에 드는 시간과 비용을 줄이기 위해 테스트 케이스를 이용하여 안전성 요구사항을 소프트웨어 개발의 가장 기본적인 검증기법인 테스팅을 이용하여 시스템이 만족하는지 확인하는 방법을 제안하고 핸드폰의 카메라 컨트롤러 예제를 이용한 사례 연구를 진행하였으며 이를 지원 위한 도구인 TC2SMV를 개발하였다. TC2SMV는 UML의 state chart diagram에서 생성된 test case들을 SMV 입력 프로그램으로 자동으로 변환하여 사용자가 SMV 모델체킹을 수행할 수 있는 환경을 제공한다.

1. 서 론

소프트웨어 시스템의 안전성을 보장하기 위해 안전성 분석(safety analysis)과 안전성 평가(safety assessment) 가 수행되고 있다. 안전성 분석은 소프트웨어 시스템이 만족해야 할 안전성 요구사항을 분석하고 개선하는 과정으로, FTA (Fault Tree Analysis) [1], FMEA (Failure Mode and Effect Analysis) [2], HAZOP (Hazard and Operability) [3]과 같은 기법들이 사용되고 있다. 안전성 평가는 안전성 분석의 결과로 나온 안전성 요구사항을 소프트웨어 시스템이 요구되는 수준까지 만족하는지 평가하는 과정으로, 안전성 분석을 언제 끝내야 하는지를 결정해 주는 역할을 한다. 따라서 신속한 안전성 평가는 비용 효율적인 (cost-effective) 소프트웨어 개발의 중요한 요소라고 할 수 있다. 그러나 안전성 분석과 평가는 전문가가 자신의 경험과 지식을 가지고 직접 수행하며 많은 시간과 비용이 든다는 문제점을 가지고 있다.

본 논문에서는 안전성 평가에 드는 시간과 비용을 줄이기 위해서 소프트웨어 개발 단계의 가장 기본적인 검증 기법인 테스팅을 이용하여 우회적으로 안전성 평가를 하는 방법을 제안한다. 안전성 요구사항과 테스팅 요구사항은 소프트웨어 시스템의 기능적 요구사항과 디자인 모델에서부터 도출되기 시작하며, 이로 인해 각 요구사항은 공통된 부분을 가질 수 있다. 이를 고려하여 테스트 케이스에 포함되는 안전성 요구사항을 확인해 줄 수 있다면 테스팅을 수행하는 것만으로 해당 요구사항들을 우회적으로 소프트웨어 시스템이 만족하고 있는지 확인 할 수 있을 것이다.

사례연구에서는 UML로 작성된 핸드폰 사진기 컨트롤러의 예제 [4]를 사용하였다. 테스트 케이스는 state chart diagram에서 자동으로 생성되었으며 [5] 테스트 케이스에 FTA에서 추출한 안전성 요구사항이 포함되어 있는지 확인하기 위해 모델체킹 기법인 SMV symbolic model checking [6]을 이용하였다. 테스트 케이스는 SMV 입력 프로그램으로 변환하여 모델 체킹의 대상이 되었고, 안전성 요구사항은 CTL property로 변환하였다. 모델 체킹의 결과가 '*TRUE*' 이면 CTL property를 생성하는데 쓰인 안전성 요구사항이 테스트 케이스에 포함 된다고 할 수 있으며 해당 테스트 케이스로 테스팅을 수행함으로써 소프트웨어 시스템이 안전성 요구사항을 만족하고 있는지 확인할 수 있다.

2. 관련 연구

Dependability는 critical 소프트웨어 시스템들이 만족해야 할 중요한 품질속성으로서 safety, security, reliability, availability의 중요한 세부 관점을 포함하고 있다 [7]. 오늘날의 인증 체제나 표준들은 테스팅에 매우 의존하고 있으나 시스템이 높은 레벨의 dependability를 보이기에 기존의 테스팅은 충분하지 않다 [8]. 시스템이 만족해야 하는 속성을 추출하고 이를 만족하는지 확인하는 통용되고 있으며, [9] 분석된 속성들을 기반으로 시스템을 구현하거나 [10, 11, 12, 13, 14] 시스템을 평가하는 연구들 [15, 16, 17]이 있었다. 한가지 예를 들면, [15]는 mechanical software fault tree를 이용하여 정형 기법을 사용한 검증에 사용하고 있다. Mechanical software fault tree는 명세나 모델, 소스 코드의 정보를 가지고 있다는 특징을 고려하여 시스템의 모델로 사용하였고 VIS 일치성 검사 [18]를 이용하여 안전성

요구사항과 mechanical fault tree의 일치 여부를 검사하여 시스템이 안전성 요구사항을 만족하는지 확인하는데 사용하였다.

3. TC2SMV

TC2SMV는 그림 1과 같이 XML 파일로 정의되어 있는 UML의 state chart diagram에서 생성한 테스트 케이스를 입력으로 받아 SMV 입력 프로그램으로 변환한다. 그리고 사용자에게 CTL property를 입력 받아 cadence SMV [19]를 실행시켜 테스트 케이스 모델이 CTL property를 만족하는지 확인 할 수 있도록 해준다. 모델체킹에 사용되는 모델은 오토마타 형태로 구성되어 있으며 테스트 케이스들을 SMV의 모델로 변환하기 위해서는 테스트 케이스들 간의 관계를 파악하여 오토마타 형태로 만들 필요가 있다. State chart diagram에서 생성된 테스트 케이스들은 표 1과 같이 state chart diagram의 state와 event/condition으로 구성되어 시스템의 상태 변화를 확인하는 형태로 되어 있기 때문에 이를 오토마타의 state와 event로 변환하고 SMV 입력 프로그램을 생성하였다.

Test Cases (Input)	Excepted output
(state = Preview, startSanpshot = 1)	(state = Snapshot)
(state = Preview, startRecord = 1)	(state = Recording)
(state = Preview, startPosteview = 1)	(state = Postview)
(state = Preview, stopPreveiw = 1, isTimeOut = 1) or (state = Preview, stopPreview = 1, isGotCameraStopEvent = 1)	(state = Stopped)
(state = Snapshot, restartpreview = 1, isGotCamerasensorError = 1) or (state = Snapshot, restartpreview = 1, isSavephoto = OK)	(state = Preview)
(state = Postview, stopPostview = 1)	(state = Preview)
(state = Stopped, exitCamera = 1)	(state = Idle)

표 1 State chart diagram에서 생성된 테스트 케이스들(발췌됨)

그림 2와 3은 각각 핸드폰 사진기 예제의 컨트롤러 요구사항 모델과 이에 대한 fault tree를 보여주고 있다. 'Preview' 상태에서 들어올 수 있는 event는 '*startPostview*', '*startRecord*', '*startSnapshot*'가 있으며 '*startSanpshot*'이 사진을 찍는 명령을 표현한 것이므로 가장 중요하다. 따라서 FTA는 사진이 찍히지 않는 경우를 failure로 설정하여 이에 대한 원인을 분석하였다. FTA의 결과인 minimal cut-set들로 추출된 안전성 요구사항은 다음과 같다.

(1) "카메라 버튼이 눌리면 세 event가 동시에 입력되더라도 '*startSanpshot*' event가 먼저 수행되어야 한다."

(2) "시스템이 사진을 찍을 준비가 되어 있을 때 사진 버튼이 눌린다면, 사진이 찍혀야 한다."

안전성 요구사항들에서 추출된 CTL property는 다음과 같다.

(1) AG(((state=Preview)&startSnapshot&startRecord &startPostview) -> AX (state=Snapshot))

(2) AG(((state=Preview)&startSnapshot&!startRecord &!startPostview) -> AF (state=Snapshot))

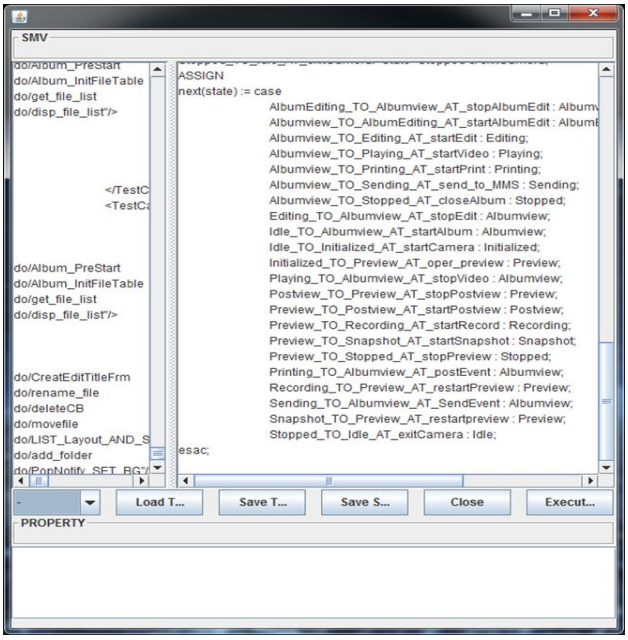

그림 1 TC2SMV의 화면 덤프

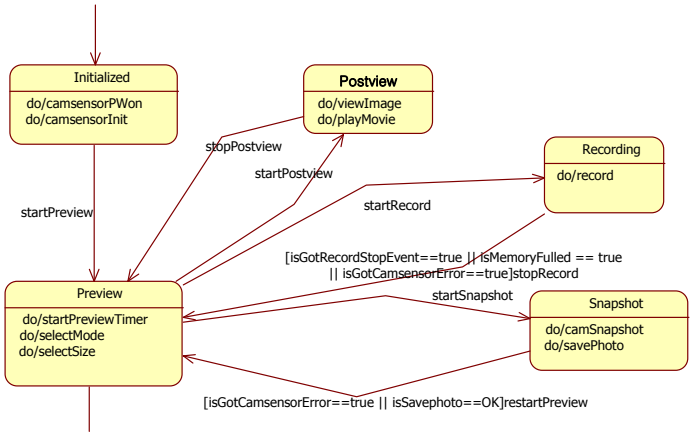

그림 2 State chart diagram으로 정의된 요구사항 모델(발췌됨)

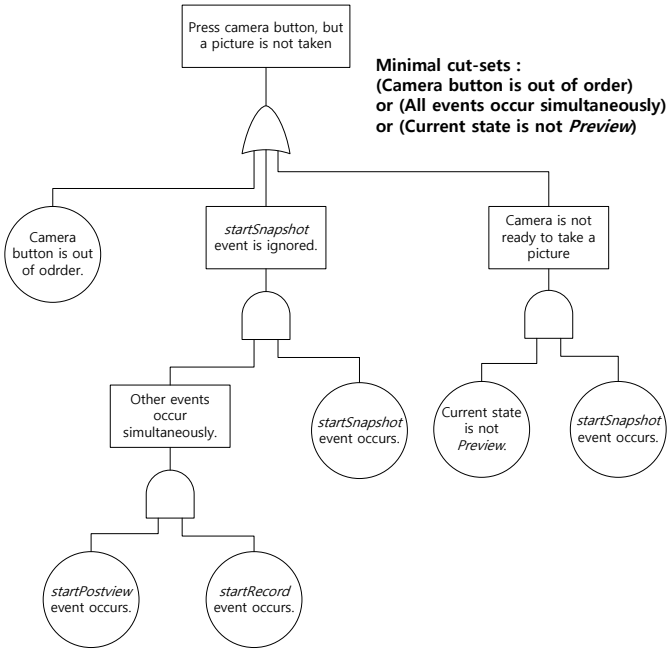

그림 3 핸드폰 컨트롤러에 대한 fault tree와 minimal cut-sets

이를 테스트 케이스를 변환한 모델이 만족하는지 모델 체킹을 수행하였다. 수행 결과 첫 번째 안전성 요구사항은 만족하지 못함을 확인하였으며, 두 번째 안전성 요구사항은 만족함을 알 수 있었다. 따라서 해당 테스트 케이스로 테스팅을 수행하여 결과가 TRUE가 나오면 면 두 번째 안전성 요구사항은 추가적인 안전성평가 기법을 적용하지 않아도 시스템이 해당 요구사항을 만족함을 확인 할 수 있다.

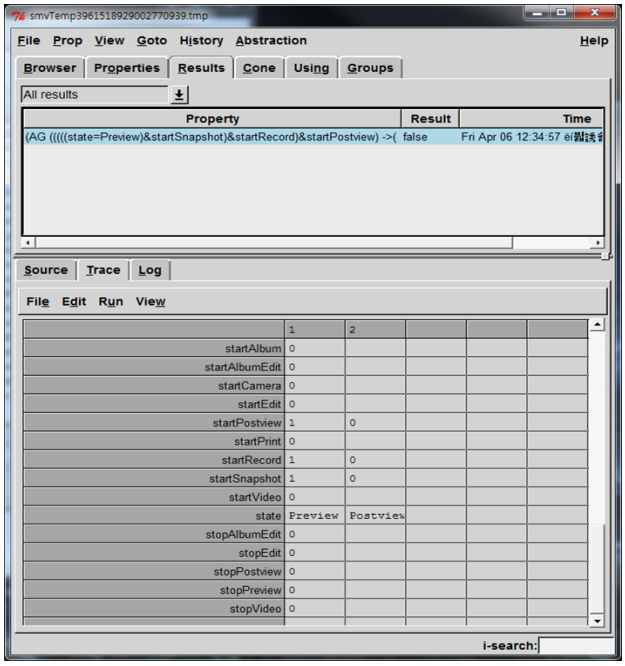

그림 4 SMV 모델 체킹의 결과와 반례

4. 결 론

본 논문에서는 안전성 평가의 시간과 비용을 줄이기 위해 테스팅을 이용한 방법을 제안하고 이를 사례 연구에 적용해 보았다. 또한 사례 연구를 통해 시스템이 안전성 요구사항을 만족하는지 확인하는 기준을 제시하였다. 연구의 기여도를 높이기 위해서는 안전성 분석과 모델 체킹 그리고 테스팅의 여러 기법들의 특징들을 파악하고 다양한 사례 연구를 통하여 각각의 특징에 맞는 세부적인 기법들을 제안해야 할 것이다.

사 사

"본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음"(NIPA-2012-(H0301-12-3004))

참고문헌

[1] W. Vesely and N. Roberts. *Fault tree handbook*. Nuclear regulatory Commission, 1987.

[2] N. Aeronautics and S. Administration. *Procedure for failure mode, effects and criticality analysis (fmeca), RM 63TMP-22*, NASA, Tech. Rep., 1966

[3] T. Kletz. *Hazop and Hazan (4th)*. IChemE, 2006.

[4] 박수진, 박수용. "임베디드 시스템을 위한 그레이-박스 기반의 소프트웨어 요구사항 명세 기법", 정보과학회 논문지: 소프트웨어 및 응용, 제 38권 9호, pp.485-490, 2011.

[5] Woo Yeol Kim, Hyun Seung Son, Robert Young Chul Kim, "A study on test case generation based on state diagram in modeling and simulation environment", Advanced Communication and Networking, Springer, 298-305, 2011.

[6] E. Clarke, K. McMillan, S. Campos, and V. Hartonas-Garmhausen. Symbolic model checking. In *Computer Aided Verification*, pages 419-422. Springer, 1996.

[7] Ian Sommerville. *SOFTWARE ENGINEERING (8th)*. Pearson College Div, 2006.

[8] Daniel Jackson, Martyn Thomas, and Lynette I. Millett. *Software for Dependable Systems: Sufficient Evidence?* The National Academy Press, 2007.

[9] D. Jackson. *A direct path to dependable software. Communications of the ACM*, 52(4):78-88, 2009.

[10] T. Kelly and R. Weaver. The goal structuring notation-a safety argument notation. In *Proc. DSN 2004 Workshop on Assurance Cases*. Citeseer, 2004.

[11] R. Lutz and A. Patterson-Hine. "Using fault modeling in safety cases". In *Software Reliability Engineering, 2008. ISSRE 2008. 19th International Symposium on*, pages 271-276. IEEE, 2008.

[12] P.J. Graydon, J.C. Knight, and E.A. Strunk. Assurance based development of critical systems. In *Dependable Systems and Networks, 2007. DSN'07. 37th Annual IEEE/IFIP International Conference on,* pages 347-357. IEEE, 2007.

[13] A. Dardenne, A. Van Lamsweerde, and S. Fickas. Goal-directed requirements acquisition. *Science of computer programming*, 20(1-2):3-50, 1993.

[14] E.S.K. Yu. Towards modelling and reasoning support for early-phase requirements engineering. In *Requirements Engineering, 1997., Proceedings of the Third IEEE International Symposium on,* pages 226-235. IEEE, 1997.

[15] Sungdeok Cha and Junbeom Yoo. Safety-focused verification using software fault trees. *Future Generation Computer Systems*, 2011. in press.

[16] E. Kang. *A framework for dependability analysis of software systems with trusted bases.* PhD thesis, Massachusetts Institute of Technology, 2010.

[17] N. Fenton, B. Littlewood, M. Neil, L. Strigini, A. Sutcliffe, and D. Wright. Assessing dependability of safety critical systems using diverse evidence. In *Software, IEE Proceedings-*, volume 145, pages 35-39. IET, 1998.

[18] Robert K. Brayton and et. al. VIS : A system for verification and synthesis. In *the Eighth International Conference on Computer Aided Verification, CAV '96,* pages 428-432, 1996.

[19] http://www2.tcs.ifi.lmu.de/lehre/SS08/Automat/smv/doc/smv/tutorial/