

Case Study of Model-Based Test with Smart TV

Seonyeol Lee[†] · Jung Ho Bae^{**} · Heung-Seok Chae^{***}

ABSTRACT

Smart TV manufacturers have constructed automatic Smart TV execution environment for reducing cost of Smart TV testing. However, the conventional test methods still require high cost because of manual test script written by testers. Test scripts manually written have less reusability, and then less reusable scripts cannot be used for new TV models. In this paper, we propose constructing reusable test models for Smart TV and present a model-based Smart TV test script generation method using the modes. And we apply the method to NetCast Smart TV to confirm the effectiveness of enhancing reusability.

Keywords : Model-Based Testing, Smart TV Test Script Generation, Reusable Test Model

모델 기반 스마트 TV 테스트 사례

이 선 열[†] · 배 정 호^{**} · 채 흥 석^{***}

요 약

스마트 TV 제조사들은 스마트 TV 테스트 비용을 줄이기 위하여 테스트 스크립트 자동 실행 환경을 구축하고 있다. 그러나 기존의 테스트 방법은 테스트 스크립트가 테스터에 의해 수동으로 작성되기 때문에, 테스트 스크립트의 재사용성은 높지 않다. 이는 새로운 TV 모델을 위한 테스트 스크립트가 새롭게 작성되어야 함을 의미한다. 본 논문에서는 재사용성이 높은 스마트 TV 테스트 모델 작성법을 제안하고 이를 이용한 스마트 TV 테스트 스크립트 생성 방법을 소개한다. 그리고 NetCast TV에 모델 기반 테스트 스크립트 생성 방법을 적용하여 유효성을 확인하였다.

키워드 : 모델 기반 테스트, 스마트 TV 테스트 스크립트 자동 생성, 재사용 가능 테스트 모델

1. 서 론

제조사들은 소비자들의 요구에 따라 다양한 스마트 TV 모델을 출시하고 있다. 다양한 화면 크기와 해상도를 가진 스마트 TV뿐만 아니라 모션인식 등 새로운 입력 장치를 도입한 스마트 TV를 출시한다. 또한 이용자경험을 토대로 스마트 TV 소프트웨어의 기능을 추가/수정/삭제하기도 한다.

스마트 TV 모델이 다양해지면 다양해질수록, 스마트 TV 테스트 비용은 증가한다. 모든 모델을 위한 개별적인 테스트 스크립트가 작성되어야 하며, 작성된 스크립트가 각각의 스마트 TV에서 실제로 수행되어야 하기 때문이다. 만약 스

마트 TV의 기능이 추가/수정/삭제된다면, 기존의 모든 테스트 스크립트는 이를 반영하여 수정되어야 한다.

테스트 비용을 줄이기 위하여 제조사들은 테스트 스크립트 자동 실행 환경을 구축하고 있다. 테스트 스크립트 실행은 단순하지만 많은 시간과 인력이 필요한 고비용의 작업이다[1]. 특히, 스마트 TV와 같은 복잡한 시스템의 스크립트 실행은 더 많은 비용을 필요로 한다. 제조사들은 테스트 스크립트 자동 실행 환경 구축하여 테스트 실행 비용을 줄일 수 있다.

그러나 기존의 테스트 방법은 여전히 많은 비용을 필요로 한다. 테스트 스크립트가 테스터에 의해 수동으로 작성되고 있기 때문이다. 테스터는 테스트 과업에 따라 테스트 스크립트를 작성하기 때문에 스크립트는 테스트 대상 TV에 밀접하게 관련된다. 즉, 스마트 TV 모델마다 테스트 스크립트가 새롭게 작성되어야 함을 의미한다[2].

테스트 스크립트 작성 비용을 줄이기 위하여 모델 기반 테스트 스크립트 생성 연구가 수행되어 왔다[3-5]. 모델 기반 테스트 스크립트 생성은 테스트 대상 시스템으로부터 생

※ 이 논문은 부산대학교 자유과제 학술연구비(2년)에 의하여 연구되었음.
※ 본 논문은 BK21플러스, IT기반 융합산업 창의인력양성사업단에 의하여 지원되었음.

† 준 회 원 : 부산대학교 전자전기컴퓨터공학부 박사수료

** 준 회 원 : 국방과학연구소 제1기술연구소 3부 선임연구원

*** 정 회 원 : 부산대학교 컴퓨터공학과 부교수

Manuscript Received : November 13, 2014

Accepted : December 29, 2014

* Corresponding Author : Heung-Seok Chae(hschae@pusan.ac.kr)

성된 모델을 바탕으로 테스트 스크립트를 생성하는 방법으로써 테스트 모델의 높은 가독성, 높은 이해성 및 높은 유지보수성은 테스트 스크립트 생성 비용을 감소시키는 데 도움이 될 수 있다.

그러나 다양한 모델이 빠르게 변화하는 분야에서는 테스트 모델 또한 변화에 대응하여 올바르게 수정되어야 한다. 즉, 테스트 모델이 재사용성을 충분히 고려하여 작성되어야 한다. 그러나 재사용성이 높은 테스트 모델 작성법에 관한 연구는 우리가 아는 범위에서 전무한 실정이며, 특히 스마트 TV 분야와 같이 빠른 변화가 일어나는 분야를 위해 이와 같은 연구가 필요하다.

이에 본 논문에서는 재사용성이 높은 스마트 TV 모델 작성 방법을 제안하고 이를 실제 Smart TV에 적용해보았다. 제안한 테스트 모델은 TV 모델의 2가지 구분 요인을 수용하기 위한 2계층으로 이루어져 있으며, 스마트 TV를 테스트하기 위해 필요한 구성요소를 가지고 있다. 그리고 모델로부터 생성된 테스트 스크립트를 실행 가능한 형태로 만들기 위한 테스트 어댑터[6, 7]를 설명하고 이를 적용하여 실행 가능한 테스트 스크립트 생성 방법을 설명한다.

본 논문의 구성은 아래와 같다. 2절에서는 제조사들의 스마트 TV 테스트 현황을 연구배경으로 설명한다. 3절에서는 스마트 TV에 모델 기반 테스트를 적용하기 위한 방법을 설명하고, 4절에서는 NetCast TV에 실제 적용 사례를 설명한다. 5절에서는 결론과 향후 연구 방향에 대해 설명한다.

2. 연구 배경

여러 제조사들은 스마트 TV 자동 테스트를 위하여 테스트 자동 실행 환경을 구축하고 있다. 실행 환경은 스마트 TV의 화면을 인식하여 이미지를 검색하고 이를 선택하여 테스트 기능을 수행한다. 또한, 테스트 기능 수행 후의 실행 결과를 예상 결과와 비교하여 테스트 기능을 자동으로 검증할 수도 있다.

그러나 자동 실행 환경의 입력인 테스트 스크립트는 테스트에 의해 수동으로 작성되어야 한다. 테스트는 테스트 대

상 기능이 테스트 환경에서 실행될 수 있도록 실행 환경이 선택할 이미지 및 행동을 테스트 스크립트로 작성해야 하고, 테스트 결과의 검증을 위한 예상 결과도 미리 정의해야 한다.

Fig. 1은 기존의 스마트 TV 테스트 방법을 나타낸다. 테스트는 요구사항으로부터 테스트 스크립트를 수동으로 작성하고, 작성된 테스트 스크립트는 테스트 자동 실행 환경에 입력되어 스마트 TV를 자동 실행한다.

수동 테스트 스크립트 작성의 문제점은 테스트가 자신의 과업 대상 모델만을 위한 테스트 스크립트를 작성하기 때문에 테스트 스크립트가 테스트 대상 모델에 매우 밀접하게 관련되어 있다는 것이다. 이는 곧, 기존의 테스트 스크립트가 다른 모델에는 재사용되기 어려우며, 새로운 TV 모델을 위해 새로운 테스트 스크립트가 작성되어야 함을 의미한다. Fig. 2는 테스트 대상 스마트 TV에 따라 달리 작성되어야 하는 테스트 스크립트를 나타낸다.

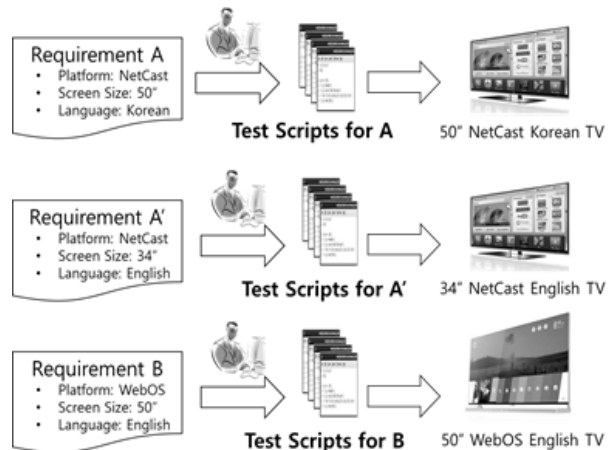


Fig. 2. Test scripts newly generated by testers according to change of Smart TV model

Fig. 2와 같이 테스트는 동일한 플랫폼을 가지지만 화면의 크기나 적용 언어가 다른 TV 모델을 위한 테스트 스크립트를 개별적으로 작성해야 한다. 그리고 NetCast TV와 WebOS TV같이 다른 플랫폼을 사용하여 요구사항이 다른 TV 모델을 위한 테스트 스크립트도 작성해야 한다.

만약 스마트 TV 테스트 스크립트 생성을 위해 모델 기반 테스트 스크립트 생성 방법을 적용한다면, 테스트 생성 비용을 줄일 수 있을 것이다. 특히, 재사용성이 높은 스마트 TV 테스트 모델을 작성한다면, 테스트 스크립트 생성 비용은 더욱 감소시킬 수 있을 것이다.

3. 모델 기반 스마트 TV 테스트 스크립트 생성

본 절에서는 모델 기반 스마트 TV 테스트 스크립트 생성 방법에 대해 소개한다. 3.1절에서는 그 절차를 소개하고, 그 이후의 절에서는 각 절차에 대해서 자세하게 설명한다.

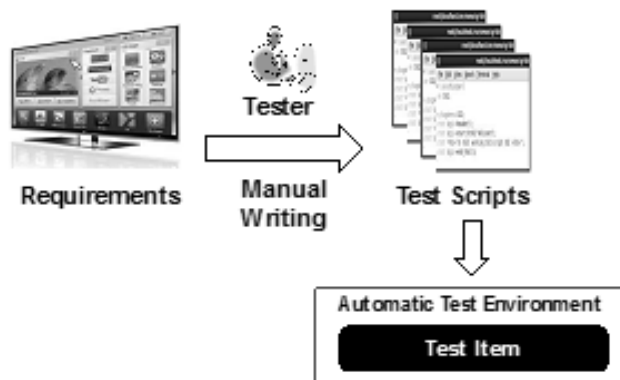


Fig. 1. Conventional Smart TV testing method

3.1 모델 기반 스마트 TV 테스트 스크립트 생성 절차

모델 기반 테스트 스크립트 생성은 시스템 요구사항과 동작에 대한 모델을 이용하여 테스트 대상 시스템의 기능적인 부분과 비기능적인 부분을 테스트하기 위한 스크립트를 생성하는 것이다[2].

본 논문에서는 모델 기반 스마트 TV 테스트 스크립트 생성을 위해 스마트 TV 테스트 모델 생성, 추상적 테스트 스크립트 생성, 테스트 어댑터 생성, 구체적 테스트 스크립트 생성의 4단계 절차를 수행하였다. Fig. 3은 테스트 스크립트 생성의 4단계 절차를 나타낸다.

❶ 테스트 모델 생성 : 스마트 TV의 요구사항에 따라 수행 기능과 예상 결과로 이루어진 테스트 모델을 생성한다. 본 논문에서는 재사용성이 높은 스마트 TV 모델로서 2계층의 테스트 모델을 제안한다.

❷ 추상적 테스트 스크립트 생성 : 테스트 모델로부터 스마트 TV 테스트 스크립트를 생성한다. 생성된 테스트 스크립트는 테스트 모델로부터 생성되었기 때문에 실제로 실행될 수 없는 형태이며, 본 논문에서는 이를 추상적 테스트 스크립트라고 정의한다.

❸ 테스트 어댑터 생성 : 테스트 모델에 기술된 기능이 테스트 실행 환경에서 실제로 수행되기 위해서는 실제 동작 코드와 대응되어야 한다. 테스트 어댑터는 테스트 모델을 테스트 환경과 대응시키기 위한 컴포넌트로, 테스트 모델의 기능을 동작 코드에 대응시킨다.

❹ 구체적 테스트 스크립트 생성 : 추상적 테스트 스크립트와 테스트 어댑터를 대응시켜 실제 스마트 TV를 실행시킬 수 있는 실행 코드를 생성한다. 이 실행 코드는 기존의 수동 테스트 스크립트와 같은 형태이며, 본 논문에서는 구체적 테스트 스크립트라고 정의한다.

3.2 테스트 모델 생성

재사용성이 높은 테스트 모델은 테스트 대상의 변경에 따른 모델의 변경이 적어야 한다. 본 절에서는 재사용성이 높은 테스트 모델을 생성하기 위한 2계층 모델을 제안하고, 이를 자세하게 설명한다.

1) 2계층 테스트 모델

본 논문에서는 스마트 TV 모델을 구분하는 변경 요인을 논리적 기능과 물리적 입력의 2가지 요인으로 가정하고, 변경 요인에 대한 재사용성이 높은 2계층의 테스트 모델을 제

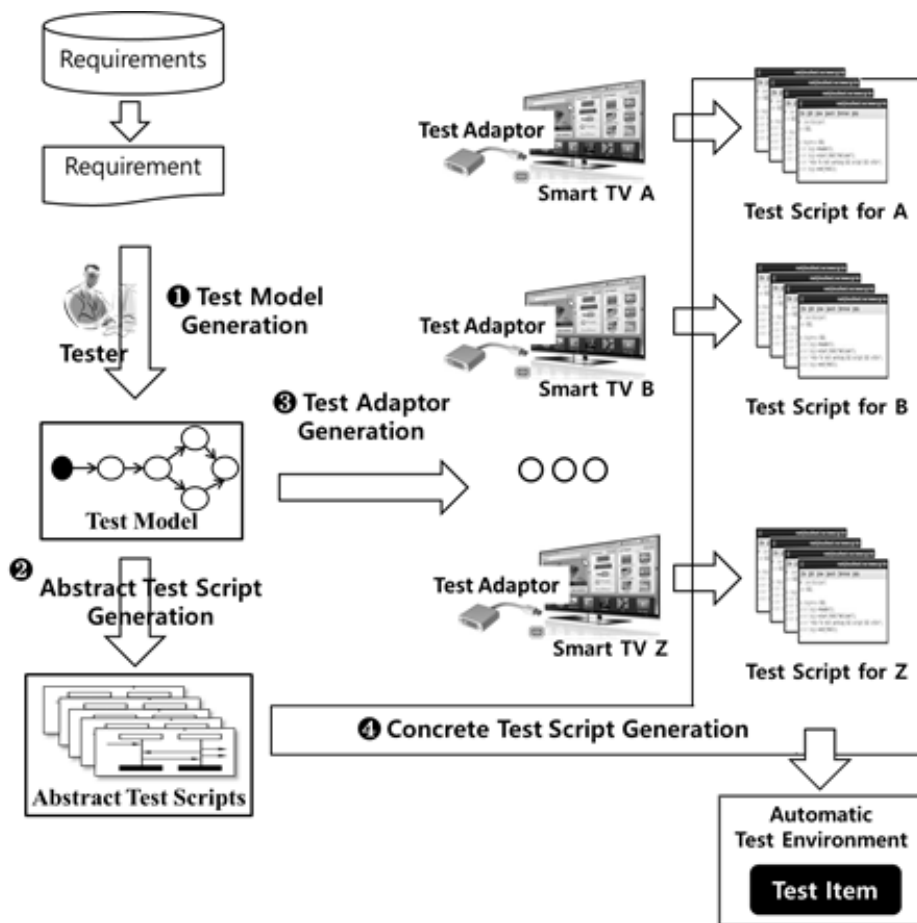


Fig. 3. The procedure for model-based Smart TV test script generation

안한다. 2가지 변경 요인을 자세하게 설명하면 아래와 같다.

① 논리적 기능 차이에 의한 구분 : 논리적 기능이란 스마트 TV가 제공하는 기능을 의미한다. 예를 들어, TV의 채널을 변경할 수 있고, 음량을 조절할 수 있다. 모바일 기기의 화면을 TV 화면을 통하여 볼 수 있으며, 콘솔 게임을 즐길 수도 있다. 위의 나열된 기능은 스마트 TV 모델을 구분할 수 있는 중요한 특징 중 하나이며, 논리적 기능의 유무에 따라 스마트 TV의 모델이 구분할 수 있다.

② 물리적 입력 차이에 의한 구분 : 물리적 입력이란 논리적 기능을 수행하기 위한 실제 입력 방법을 의미한다. 예를 들어, TV 채널을 변경하기 위해서는 일반 리모컨의 채널 변경 버튼을 누를 수도 있고, 매직 리모컨의 채널 변경 버튼을 누를 수도 있다. 또한, 음성이나 모션의 인식을 통하여 채널 변경을 수행할 수도 있다. 위의 나열된 입력의 지원 유무는 스마트 TV 모델을 구분할 수 있는 중요한 요인이 될 수 있다.

본 논문에서는 2가지 변경 요인에 대해 재사용성이 높은 테스트 모델을 생성하기 위해 UI 논리적 모델(UI Logical Model)과 UI 물리적 모델(UI Physical Model)의 2계층으로 구성된 테스트 모델을 제안한다. Fig. 4는 제안한 2계층의 테스트 모델을 나타낸다.

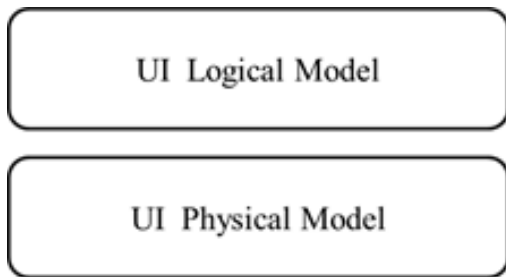


Fig. 4. 2-Layered Smart TV Test Model

UI 논리적 모델은 스마트 TV 기능에 대한 변경을 수용하는 계층이고, UI 물리적 모델은 물리적 입력에 대한 변경을 수용하는 계층이다. 기능과 입력에 대한 계층을 구분함으로써, 입력 변경에 따른 테스트 모델의 변경을 최소화하고자 한다.

2) UI 논리적 모델

UI 논리적 모델은 테스트 대상 시스템의 테스트 기능을 기술하기 위한 모델이다. 테스트 기능은 수용 가능한 입력 기능의 흐름과 그에 따른 예상 출력으로 표현될 수 있다. 이는 상태 머신과 유사하게 표현될 수 있으며, 본 논문에서는 UML 2.0의 프로토콜 상태 머신 다이어그램 표기법을 토대로 UI 논리적 모델을 기술한다. Fig. 5는 UI 논리적 모델의 주요 구성 요소인 상태와 전이의 표기법을 나타낸다.

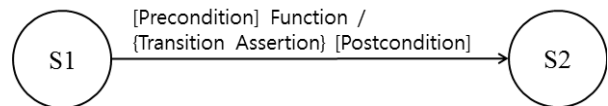


Fig. 5. State and Transition Notation of UI Logical Model

UI 논리적 모델은 Fig. 5와 같이 상태와 전이로 표현된다. 상태는 스마트 TV의 화면 나타내며, 전이는 화면 간 이동을 나타낸다. 전이는 화면 이동을 일으키는 사건인 기능(Function)과 기능이 발생하기 전에 만족해야 할 선행조건(Precondition), 화면 이동 중 만족해야 할 전이 중 조건(Transition Assertion), 그리고 기능이 수행된 후 만족해야 할 후행조건(Postcondition)으로 구성된다.

전이 중 조건은 전이가 수행되는 과정에서 만족되어야 하는 조건을 나타내며, 스마트 TV의 테스트 모델을 작성하기 위해서 본 논문에서 새롭게 추가한 전이의 구성 요소이다. 전이 중에 만족되어야 하는 스마트 TV의 상태뿐만 아니라 TV 화면에 표현되어야 애니메이션의 품질 조건 등을 기술하기 위해 추가하였다. Fig. 6은 UI 논리적 모델의 기술을 위한 간단한 예제이다.

Fig. 6은 NetCast 스마트 TV의 영상 화면에서 홈 화면으로의 이동을 나타내기 위한 UI 기능 모델이다. Fig. 6의 스마트 TV는 초기 상태(initial state)가 가리키는 TV 영상 화면에서 시작한다. TV 영상 화면은 home() 이벤트에 의해 홈 화면으로 이동할 수 있다. 단, 홈 화면으로 이동하기 위해서는 TVMode = DTV인 선행조건을 만족해야 한다. 화면 이동이 진행 중일 때, 스마트 TV는 ImageMode = Normal && SoundMode = Normal을 만족해야 한다. 화면 이동이

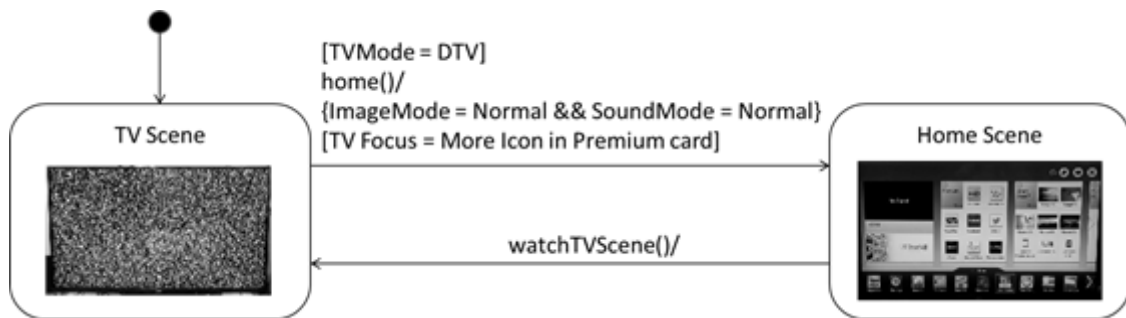


Fig. 6. An example of UI Logical Model

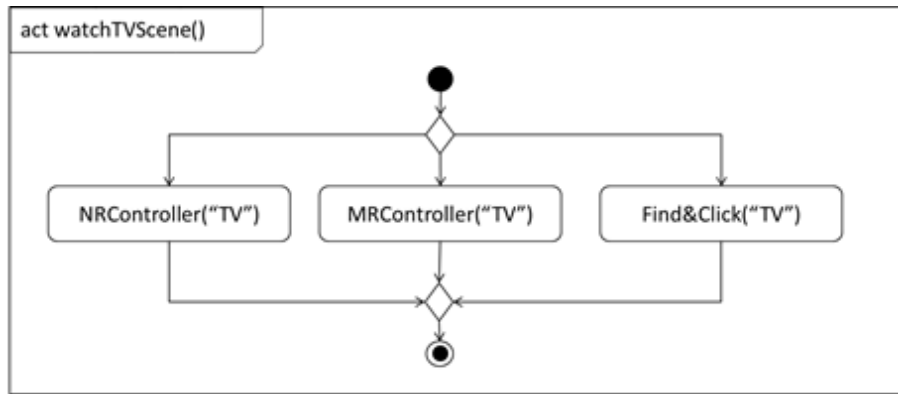


Fig. 7. An example of UI Physical Model

완료되면 TVFocus = More Icon in Premium card 조건을 만족해야 한다.

3) UI 물리적 모델

UI 물리적 모델은 UI 논리적 모델의 기능에 대한 입력 방법을 기술한다. 입력 방법은 실제 수행되는 명령의 순서가 중요하기 때문에 UML 2.0의 활동 다이어그램(Activity Diagram)의 표기법을 따른다. Fig. 7은 Fig. 6의 watchTVScene() 기능을 UI 물리적 모델로 표현한 예를 나타낸다.

Fig. 7 좌측 상단의 act는 활동 다이어그램임을 나타내는 키워드이며, watchTVScene()는 활동 다이어그램이 표현하

는 기능을 나타낸다. 모서리가 둥근 사각형은 물리적 입력을 나타내며, 활동이라고 불린다. Fig. 7의 UI 물리적 모델에 따르면 watchTVScene() 기능은 3가지 물리적 입력에 의해 수행 가능하다. 일반 리모컨이나 매직 리모컨의 TV 버튼을 누를 수 있으며, 화면의 TV 이동 버튼을 선택할 수도 있다.

3.3 추상적 테스트 스크립트 생성

추상적 테스트 스크립트는 UI 논리적 모델로부터 생성된 테스트 케이스를 의미한다. UI 논리적 모델은 기본적으로 프로토콜 상태 머신 다이어그램의 표기법을 따르기 때문에, 추상적 테스트 스크립트는 상태 커버리지와 같은 상태 머신

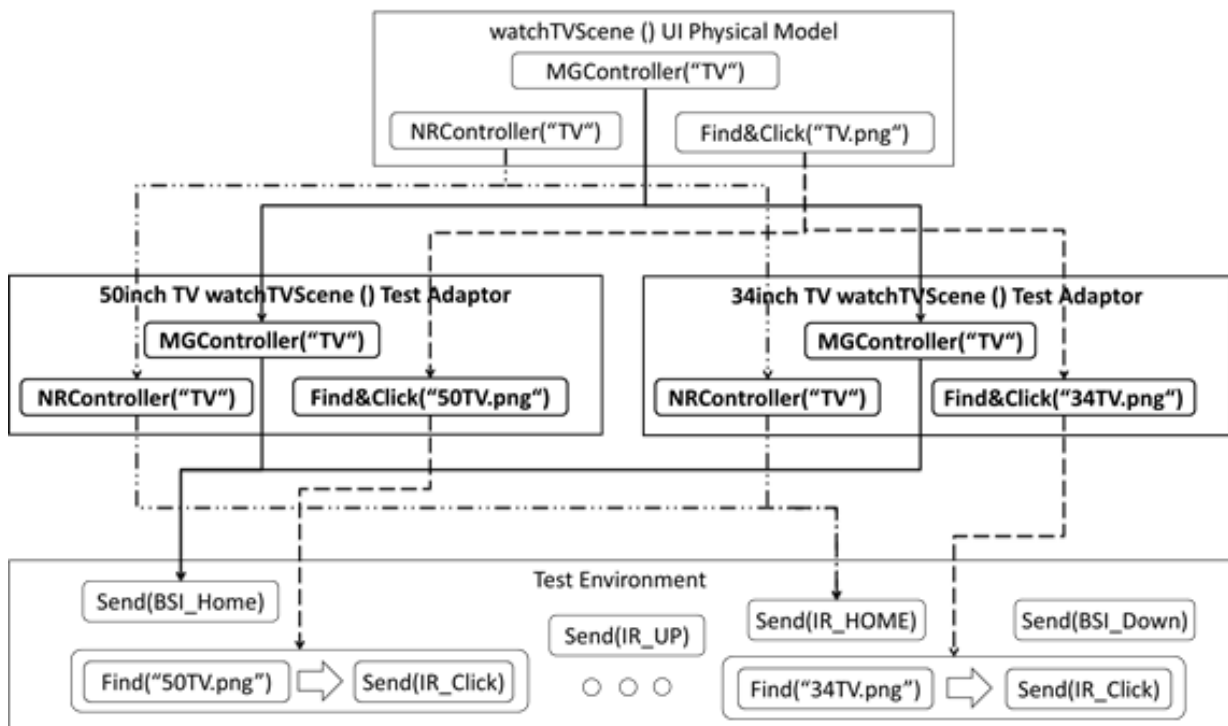


Fig. 8. Mapping activities of UI Physical Model to execution code by test adaptors

테스트 케이스 생성 방법에 의해 생성될 수 있다[8]. Table 1은 Fig. 6의 모델에 상태 커버리지를 적용하여 생성한 추상적 테스트 스크립트를 나타낸다.

Table 1. Ab abstract test script derived from the UI Logical Model

No.	Abstract Test Script
1	TV Scene $\xrightarrow{\text{home()}}$ Home Scene $\xrightarrow{\text{watchTVScene()}}$ TV Scene

Table 1의 1번 테스트 스크립트는 TV Scene 화면에서 home() 기능을 수행하면, Home Scene 화면으로 이동하고, Home Scene 화면에서 homeExit() 기능을 수행하면 다시 TV Scene 화면이 되는 것을 의미한다. 선행조건과 같은 전이의 모든 구성요소도 추상적 테스트 스크립트에 포함되어야 하지만, 간단한 표기를 위해 기능을 제외한 나머지 요소들은 제외하였다.

3.4 테스트 어댑터 생성

테스트 어댑터는 UI 물리적 모델의 활동과 테스트 환경에 구현되어 있는 TV 동작 코드를 대응시키는 컴포넌트이다. TV 동작 코드는 리모컨의 채널 변경 명령과 같이 스마트 TV를 실제로 동작시키는 시그널을 의미하며, 시그널의 형태에 따라 이미지, 음성 등의 매개변수를 가질 수 있다. 매개변수를 가지는 동작 코드와 UI 물리적 모델의 활동을 대응시키기 위해서 테스트 어댑터 또한 매개변수를 가져야 한다.

테스트 어댑터는 UI 물리적 모델의 활동을 다양한 TV 모델을 위한 실행 환경의 동작 코드로 대응시킬 수 있다. 예를 들어, NetCast 플랫폼의 스마트 TV가 Fig. 6과 Fig. 7의 테스트 모델을 가진다고 가정해 보자. NetCast TV는 50인치, 34인치 등 다양한 화면 크기의 TV 모델이 있으며, 각 모델의 UI 물리적 모델의 활동은 테스트 어댑터에 의

해 실행 환경의 동작 코드와 대응될 수 있다. Fig. 8은 50인치 TV와 34인치 TV를 위한 테스트 어댑터에 의해 UI 물리적 모델과 테스트 환경의 동작 코드가 대응되는 것을 나타낸다.

Fig. 8의 NRController(“TV”)와 MRController(“TV”) 활동은 50인치와 34인치 TV에서 동일한 리모컨 입력이기 때문에 테스트 환경에서도 같은 동작 코드를 가지며, 50인치와 34인치 TV 테스트 어댑터에 의해서 Send(IR_HOME)과 Send(BSI_HOME) 동작 코드로 대응된 것을 확인할 수 있다. 그러나 Find&Click(“TV.jpg”) 활동은 2가지 TV의 화면 크기 차이로 인해 미리 정의되어야 하는 이미지의 크기가 다르다. 2가지 TV에서 필요한 이미지를 각각의 테스트 어댑터에 정의하여 UI 물리적 모델의 활동은 동일하지만 다르게 동작하는 동작 코드에 대응시킬 수 있다.

3.5 구체적 테스트 스크립트 생성

구체적 테스트 스크립트는 추상적 테스트 스크립트에 기술된 UI 논리적 모델의 기능을 테스트 어댑터에 의해 대응된 동작 코드로 나열하여 생성할 수 있다. Fig. 9는 Table 1의 추상적 테스트 스크립트로부터 구체적 테스트 스크립트의 생성을 나타낸다.

Fig. 9는 추상적 테스트 스크립트로부터 파이썬 코드로 작성된 구체적 테스트 스크립트의 생성을 나타낸다. UI 논리적 모델의 home() 기능이 UI 물리적 모델의 NRController(“HOME”) 활동에 의해 수행되며, 이 활동은 테스트 어댑터에 의해 Send(IR_HOME) 동작 코드로 대응된다. 이와 동일하게, watchTVScene() 기능이 Send(BSI_HOME)으로 대응된다. 대응된 동작 코드들은 추상적 테스트 스크립트의 기능 나열 순서에 의해 나열되며, 마지막으로 실행될 수 있도록 코드를 생성하여 구체적 테스트 스크립트가 완성된다. Fig. 9의 구체적 테스트 스크립트는 아래와 같이 TV를 동작시킨다.

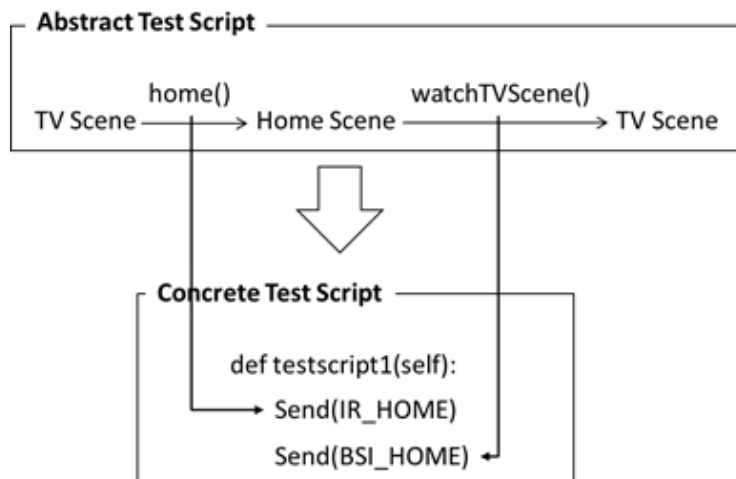


Fig. 9. A concrete test script derived from the abstract test script

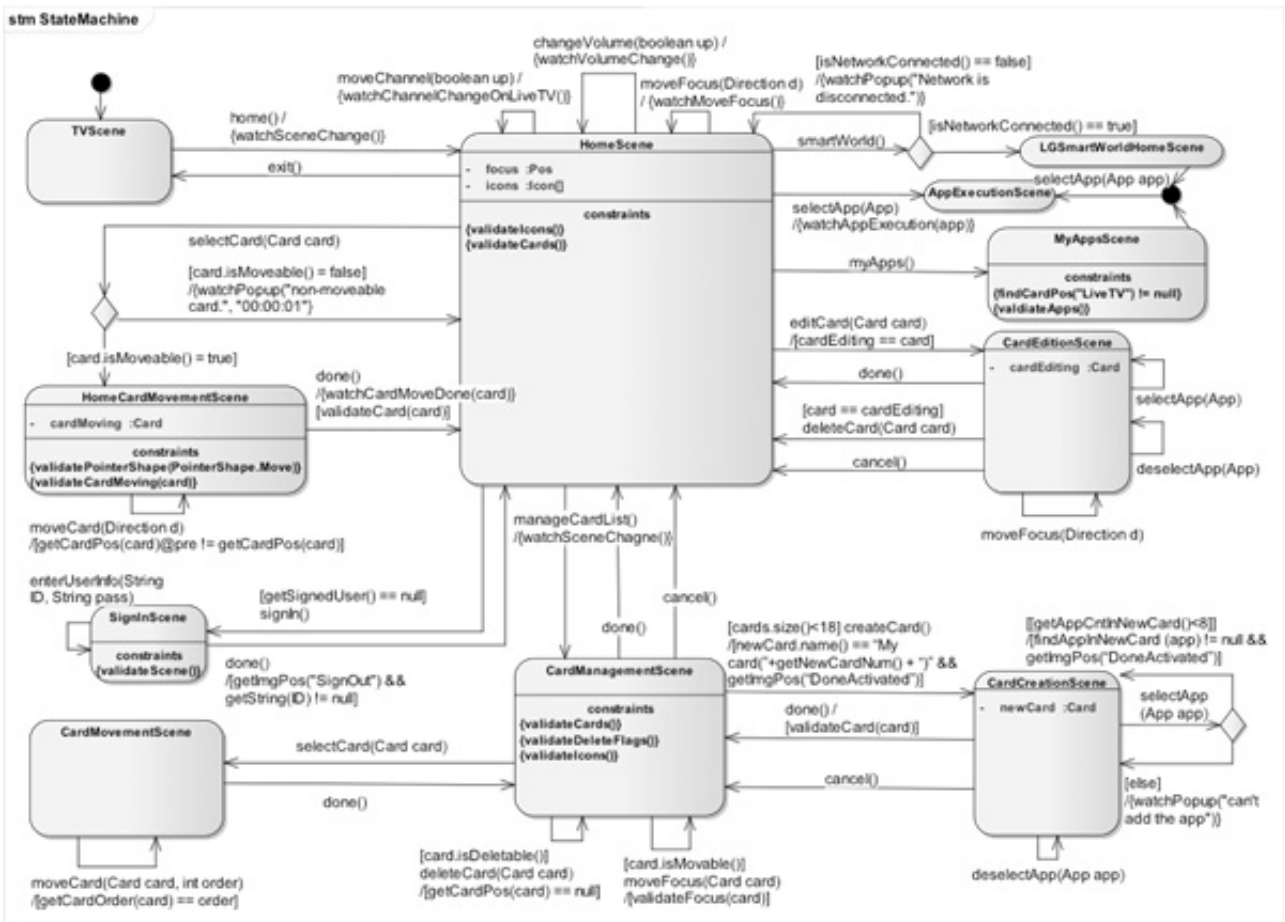


Fig. 10. NetCast Smart TV UI Logical Model

① TV 영상 화면에서 일반 리모컨의 홈 화면 이동 시그널인 IR_HOME 시그널을 발생시켜 TV를 홈화면으로 이동시킨다.

② 홈 화면에서 매직 리모컨의 TV 영상 화면 이동 시그널인 BSL_HOME 시그널을 발생시켜 TV 영상 화면으로 이동시킨다.

UI 논리적 모델의 기능에 대한 UI 물리적 모델의 활동을 다르게 조합하여 구체적 테스트 스크립트를 다양하게 생성할 수 있다. 예를 들어, Fig. 9와 달리, TV 영상 화면에서 매직 리모컨을 이용하여 홈 화면으로 이동한 후, 홈 화면에서 화면의 이미지를 선택하여 TV 영상 화면으로 이동할 수도 있다.

4. 사례 연구

본 절에서는 NetCast TV에 모델 기반 테스트 스크립트 생성 방법을 적용한 사례를 설명한다. 앞서 소개한 4단계 절차에 따라 사례 연구를 수행하였으며, 아래의 절에서 각 절차를 설명한다.

4.1 테스트 모델 생성

1) UI 논리적 모델

NetCast TV의 사용자들은 홈 화면에서 스마트 TV 앱을 관리할 수 있다. 특히, 카드라는 개념을 도입하여 사용자의 의도에 맞게 앱들을 분류할 수 있다. Fig. 10은 NetCast 스마트 TV의 카드 관리 부분에 대한 UI 논리적 모델을 나타낸다.

스마트 TV는 홈 화면으로 이동하여 스마트 TV 앱을 실행시킬 수 있다. 또한 카드를 생성, 수정, 삭제, 이동하여 사용자가 더욱 편하게 스마트 TV를 사용할 수도 있도록 한다. 이를 표현하기 위하여 Fig. 10은 11가지 화면과 18가지 기능을 사용하였다. Table 2와 Table 3은 사용된 화면과 기능을 나타낸다.

2) UI 물리적 모델

Fig. 10의 논리적 UI 모델에는 총 20개의 기능이 존재한다. 본 논문에서는 지면 제약으로 인하여 4가지 기능에 대한 UI 물리적 모델을 소개한다. Fig. 11은 NetCast Smart TV에 대한 4가지 UI 물리적 모델은 나타낸다.

Table 2. The states specified in Fig. 10 UI Logical Model

State name	Description
TV Scene	Scene for watching TV
Home Scene	Smart TV Home Scene
SmartWorld HomeScene	Smart World Home Scene
AppExecution Scene	Scene of a executing application
MyAppScene	Scene of listing apps selected by users
CardEdition Scene	Scene for editing a group of apps
HomeCard MovementScene	Scene for moving cards in TV Home
SignInScene	Login Scene
CardMovement Scene	Scene for moving cards
CardManangement Scene	Scene for managing cards
CardCreation Scene	Scene for creating a new card

Fig. 11의 상단home() 기능은 앞서 설명한 watchTVScene() 기능과 유사한 UI 물리적 모델을 가진다. home() UI 물리적 모델은 일반 리모컨, 매직 리모컨 그리고 화면의 홈 이동 버튼을 통하여 TV 영상 화면에서 홈 화면으로 이동할 수 있음을 나타낸다. home() 모델을 제외한 3가지 모델은 홈

화면에서 이미지를 선택하여 실행하는 물리적 입력만을 가지기 때문에 Find&Click() 활동만이 존재한다. 그러나 3가지 활동은 화면에서 선택해야 하는 이미지가 다르기 때문에 각기 다른 매개변수를 가진다. 이 중 selectApp() 모델은 테스트할 앱의 이미지를 모델의 매개변수로 정의함으로써, 테스트 앱의 변화에 유연하게 대처할 수 있도록 하였다.

4.2 논리적 테스트 스크립트 생성

논리적 테스트 스크립트는 UI 논리적 모델의 탐색을 통하여 생성된다. Fig. 12에 표시된 붉은선은 UI 논리적 모델의 탐색 방법을 나타낸다.

Fig. 12의 붉은선은 새로운 카드를 생성하는 테스트 시나리오를 나타낸다. TV 영상 화면인 TVScene 상태에서 home() 기능을 수행하면 스마트 TV 화면은 홈 화면을 나타내는 HomeScene으로 이동한다. HomeScene 화면에서 카드를 관리하기 위해 manageCardList() 기능을 수행하면 카드 관리를 위한 화면인 CardManagementScene 화면에 도달한다. CardManagementScene 화면에서 카드를 생성하기 위한 createCard() 기능을 수행하면 카드 생성 화면인 CardCreationScene 화면으로 전환된다. CardCreationScene 화면에서 카드에 포함될 앱을 selectApp(App app) 기능으로 선택하고 완료를 의미하는 done() 기능을 수행하여 카드 생성을 완료한다. Fig. 13은 언급된 테스트 시나리오를 논리적 테스트 스크립트로 표현한 것이다.

Table 3. The functions specified in Fig. 10 UI Logical Model

No.	Category	UI Event	Description
1	General	home()	Change from TV Scene to Home Scene
2		moveChannel(boolean up)	Change TV channel
3		changeVolume(boolean up)	Change TV volume
4		moveFocus(Direction d)	Move TV focus
5		smartWorld()	Change from Home Scene to Smart World Scene
6		selectApp(App app)	Select TV App
7	Scene Transition	myApps()	Change from Home Scene to My App Scene
8		editCard(Card card)	Change from Home Scene to Card Edit Scene
9		manageCardList()	Change from Home Scene to Card Management Scene
10		signIn()	Change from Home Scene to Login Scene
11		selectCard(Card card)	Select a card
12		exit()	Change from Home Scene to TV Scene
13	Card	done()	Complete a current task
14		deleteCard(Card card)	Delete a card
15		cancel()	Cancel a current task
16		deselectApp(App app)	Except a app from a card
17	App	selectApp(App app)	Add a app to a card
18		createCard()	Change from Home Scene to Card Creation Scene
19		moveCard(Card card, int order)	Move card location
20	Sign	enterUserInfo(String ID, String pass)	Input Login Information

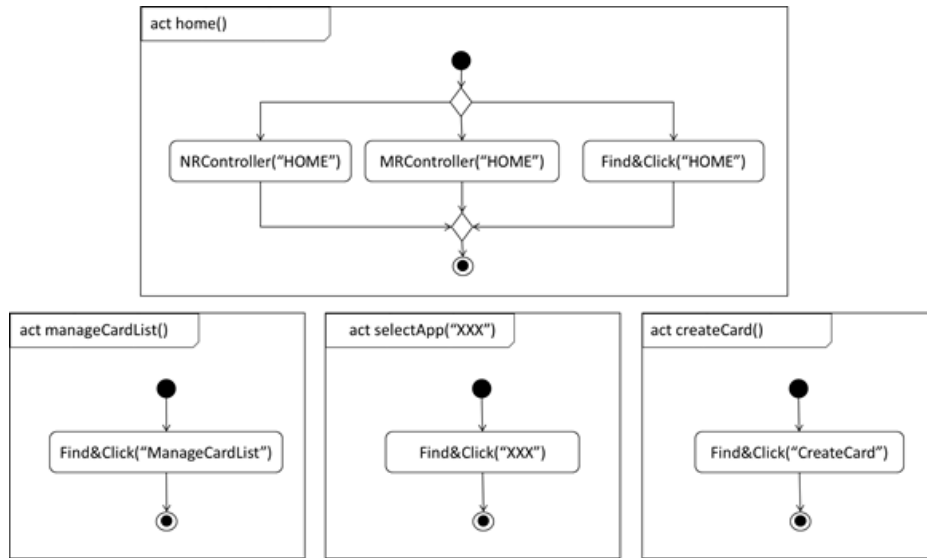


Fig. 11. Four UI Physical Models of NetCast Smart TV

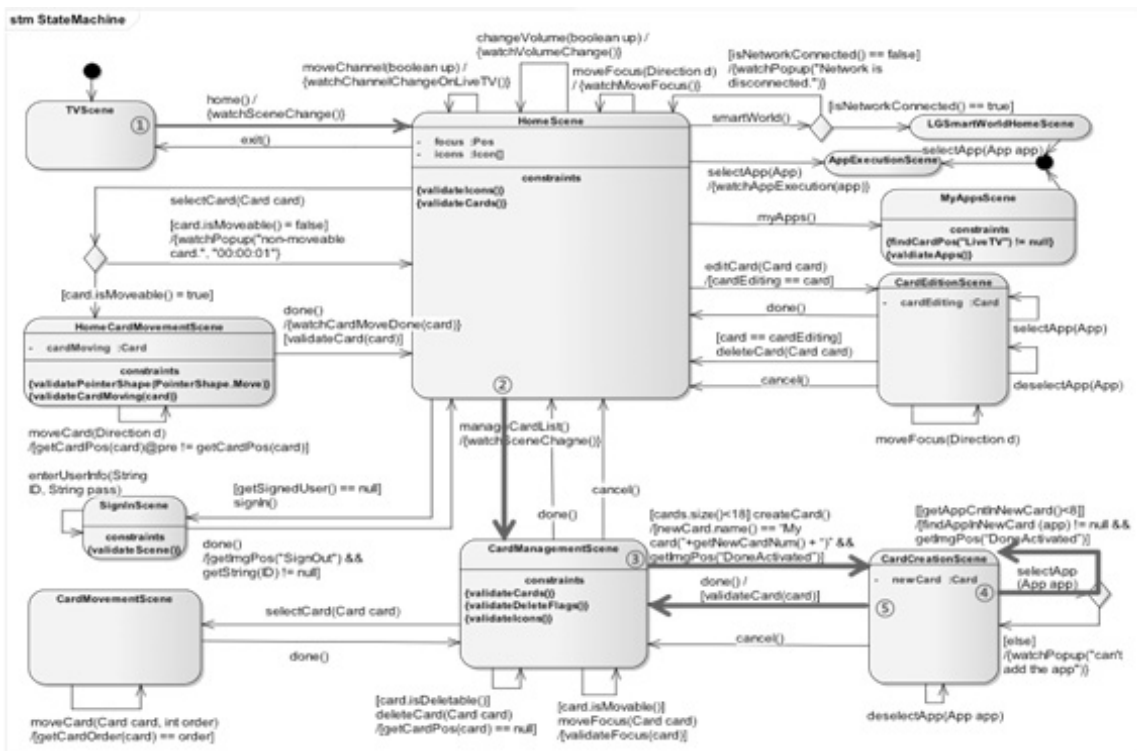


Fig. 12. Abstract test script creation by searching UI Logical Model

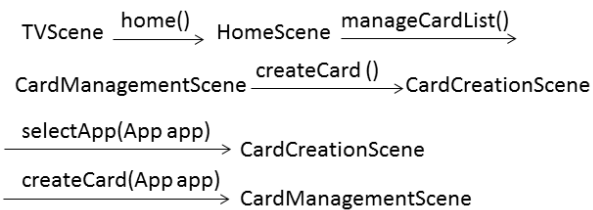


Fig. 13. Card creation abstract test script

4.3 테스트 어댑터 생성

NetCast TV의 테스트 어댑터는 UI 논리적 모델의 20가지 기능에 대한 모든 활동을 UI 동작 코드에 대응시켜야 한다. 그러나 지면 관계 상, 본 절에서는 Fig. 11의 6가지 활동을 동작 코드로 대응시키는 테스트 어댑터에 대해 설명한다. Table 4는 Fig. 11의 활동과 실행 환경의 동작 코드를 대응시키는 테스트 어댑터를 나타낸다.

Table 4. Test Adopters which correspond UI Physical Models of NetCast TV to Execution Codes


Activity	NetCast Test Adopter	Execution Code
NRCntroller (“HOME”)	NRCntroller (“HOME”)	Send(IR_HOME)
MRCntroller (“HOME”)	MRCntroller (“HOME”)	Send(BSL_HOME)
Find&Click (“HOME”)	Find&Click (“home.png”)	Find(“home.png”) Send(IR_Click)
Find&Click (“MngCards”)	Find&Click (“MngCards.png”)	Find(“MngCards.png”) Send(IR_Click)
Find&Click (“XXX”)	Find&Click (“XXX”)	Find(“XXX”) Send(IR_Click)
Find&Click (“CrtCard”)	Find&Click (“CrtCard.png”)	Find(“CrtCard.png”) Send(IR_Click)

Table 4의 테스트 어댑터는 NRCntroller(“HOME”)과 MRCntroller(“HOME”)과 같이 하나의 활동을 하나의 동작 코드로 대응시킬 수도 있으며, Find&Click() 활동과 같이 여러 동작 코드로 대응시킬 수도 있다. 여러 동작 코드로 대응시킬 경우에 동작 코드는 수행 순서를 가진다.

4.4 물리적 테스트 스크립트 생성

Fig. 13의 논리적 테스트 스크립트와 Table 4의 테스트 어댑터를 대응시켜 구체적 테스트 스크립트를 생성할 수 있다. Fig. 14는 추상적 테스트 스크립트로부터 카드 생성 구체적 테스트스크립트의 생성을 나타낸다.

추상적 테스트 스크립트의 기능을 테스트 어댑터의 대응에 의한 동작 코드로 치환하여 구체적 테스트 스크립트를 생성할 수 있다. 그러나 selectApp(App app) 기능의 테스트 어댑터는 매개변수를 가지고 있기 때문에 매개변수가 정의되어야 한다. Fig. 14에서는 이 매개변수를 YouTube 앱으로 정의하여 YouTube 앱을 카드에 추가하는 구체적 테스트 스크립트를 작성하였다. Fig. 14의 구체적 테스트 스크립트는 아래의 절차에 따라 NetCast 스마트 TV를 테스트한다.

- ① Send(IR_HOME) : 스마트 TV 홈 화면으로 이동한다.
- ② Find(“ManageCardList.png”) : 홈 화면에의  위치를 찾고, TV 커서를 이동시킨다.
- ③ Send(IR_Click) : TV 커서를 클릭하여, 카드 편집 화면으로 이동한다.

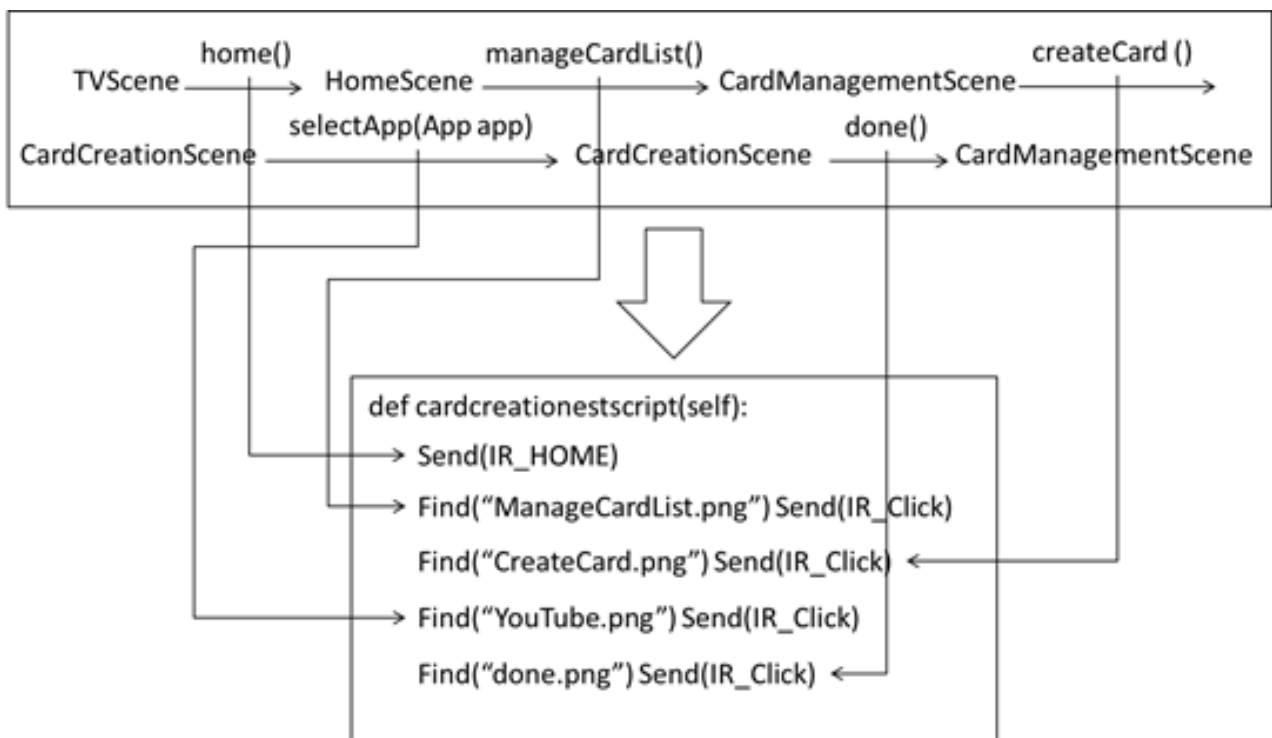


Fig. 14. Card Creation Concrete Test Script Generation

- ④ Find("CreateCard.png") : 카드 편집 화면에서  위치를 찾고, 커서를 이동시킨다.
- ⑤ Send(IR_Click) : TV 커서를 클릭하여, 카드 선택 화면으로 이동한다.
- ⑥ Find("YouTube.png") : 카드 생성 화면에서  위치를 찾고, 커서를 이동시킨다.
- ⑦ Send(IR_Click) : TV 커서를 클릭하여, YouTube 앱을 선택한다.
- ⑧ Find(done.png) : YouTube 앱이 선택된 상황에서  위치를 찾고 커서를 이동시킨다.
- ⑨ Send(IR_Click) : TV 커서를 클릭하여, 카드 생성을 완료한다.

5. 결론 및 향후 연구

기존의 스마트 TV 테스트는 테스터에 의해 테스트 스크립트를 수동으로 작성하고 테스트 자동 실행 환경이 스크립트를 자동으로 실행하여 수행된다. 스크립트의 자동 실행은 스마트 TV의 테스트 비용을 줄일 수 있었지만, 수동으로 작성되는 스크립트는 재사용이 낮기 때문에 새로운 스크립트의 작성에 여전히 많은 비용을 필요로 한다.

본 논문에서는 재사용성이 높은 스마트 TV 테스트 모델을 이용하여 스마트 TV 테스트 스크립트를 생성하고자 하였다. 이를 위하여 2계층의 스마트 TV 테스트 모델 작성 방법을 제안하였으며, 테스트 모델로부터 생성된 테스트 스크립트를 실행하기 위한 테스트 어댑터 적용 방법을 설명하였다. 그리고 테스트 모델의 재사용성 가능성을 확인하기 위해 NetCast TV에 모델 기반 테스트 스크립트 생성 방법을 적용하였다. 향후 연구에서는 기존 테스트 생성 방법의 재사용성 대비 제안된 방법의 재사용성 향상에 관한 비교 연구를 수행하겠다. 그리고 제안된 방법을 이용한 스마트 TV의 실제 테스트 자동화에 관한 연구를 수행하겠다.

References

[1] M. Utting and B. Legeard, Practical Model-Based Testing, Morgan Kaufmann, 2010.
 [2] J. Zander, I. Schieferdecker, and P. J. Mosterman, Model-Based Testing for Embedded Systems, CRC Press, 2011.

[3] W. L. Andrade and P. D. L. Machado, "Generating Test Cases for Real-Time Systems Based on Symbolic Models," Software Engineering, IEEE Transactions on, Vol.39, No.9, pp.1216-1229, 2013.
 [4] S. Mohalik, A. A. Gadkari, and A. Yeolekar, "Automatic test case generation from Simulink/Stateflow models using model checking," Softw. Test. Verif. Reliab., 2014.
 [5] P. Fröhlich and J. Link, "Automated Test Case Generation from Dynamic Models," Presented at the ECOOP '00: Proceedings of the 14th European Conference on Object-Oriented Programming, 2000.
 [6] E. Farchi, A. Hartman, and S. S. Pinter, "Using a model-based test generator to test for standard conformance," IBM Systems Journal, Vol.41, No.1, pp.89-110, 2002.
 [7] A. Hartman and K. Nagin, "The AGEDIS tools for model based testing," ACM SIGSOFT Software Engineering Notes, Vol.29, No.4, pp.129-132, Jul., 2004.
 [8] A. J. Offutt, Y. Xiong, and S. Liu, "Criteria for generating specification-based tests," Presented at the Engineering of Complex Computer Systems, 1999. ICECCS '99. Fifth IEEE International Conference on, pp.119-129, 1999.



이 선 열

e-mail : sun302412@pusan.ac.kr
 2008년 부산대학교 정보컴퓨터공학부(학사)
 2010년 부산대학교 컴퓨터공학과(석사)
 2010년~현재 부산대학교 전자전기컴퓨터공학부 박사수료
 관심분야: 모델 기반 테스트, 소프트웨어 결합 예측 등



배 정 호

e-mail : jhbae@add.re.kr
 2007년 부산대학교 정보컴퓨터공학부(학사)
 2009년 부산대학교 컴퓨터공학과(석사)
 2014년 부산대학교 컴퓨터공학과(박사)
 2014년~현재 국방과학연구소 제1기술 연구본부 3부 선임연구원
 관심분야: 모델 기반 테스트, 테스트 자동화 등



채 흥 석

e-mail : hschae@pusan.ac.kr

1996년 서울대학교 원자핵공학(학사)

1996년 한국과학기술원 전산학(석사)

2000년 한국과학기술원 전산학(박사)

2000년~2003년 (주)동양시스템즈 기술연
구소 선임연구원

2003년~2004년 한국과학기술원 전산학과 초빙교수

2004년~2010년 부산대학교 컴퓨터공학과 조교수

2010년~현 재 부산대학교 컴퓨터공학과 부교수

관심분야: 객체지향 방법론, 소프트웨어 테스팅, 소프트웨어 메트릭,
소프트웨어 유지보수, 미들웨어 설계, 프로덕트라인
공학