

Orthogonal Array와 Equivalence Class를 이용한 효율적인 Test Set 생성 기법

김은희⁰ 여기대
삼성전자 시스템연구소
{eunhee2.kim⁰, gidae.yeo}@samsung.com

Effective Test Set Selection Methodology using Orthogonal Array and Equivalence Class

Eunhee Kim⁰ Gidae Yeo
System R&D Laboratories, Samsung Electronics Co.Ltd

요 약

최근 소프트웨어가 점점 복잡해짐에 따라 다양한 상황에 대한 테스트의 중요도도 함께 증가하지만, 다양한 상황을 모두 고려하여 테스트하는 것은 많은 시간과 비용을 요구한다. 본 논문에서는 이러한 복잡한 시스템의 테스트를 위해 Orthogonal Array와 Equivalence Class 기법을 이용하여 test suite의 중복성을 낮추는 효율적인 test set을 생성하는 기법을 제안한다. 이렇게 제안된 기법을 OMA DRM(Digital Rights Management) System에 적용한 사례를 통해 기법의 우수성을 보인다.

1. 서 론

최근 소프트웨어가 점점 복잡해지고 있으며, 이에 따라 해당 소프트웨어가 가지는 다양한 상황에 대한 테스트의 중요도도 함께 증가되고 있다. 그러나, 복잡한 소프트웨어의 행동을 쉽게 이해할 수 있는 방법적 모델 수립이 어렵고, 이들을 테스트하기 위한 test suite도 복잡해지고 있다. 이러한 복잡한 소프트웨어를 채택하는 시스템을 테스트하기 위한 다양한 시도가 이루어지고 있다.

본 논문에서는 다양한 기능을 수행하는 시스템의 test suite의 중복성을 줄이고, test set의 효율성을 높이기 위해 Orthogonal Array와 Equivalence Class 기법을 이용한 test set 선정 기법을 제안한다. 본 논문에서 제안하는 테스트 데이터 선정 기법을 OMA(Open Mobile Alliance) DRM(Digital Rights Management) Spec.[1]을 기반으로 하여 본 연구진의 개발팀이 개발한 OMA DRM System의 시스템 테스트에 적용한 사례를 제시한다.

본 논문의 2장은 Orthogonal Array Testing과 Equivalence Class Testing 기법 및 용어에 대해 기술하고, 3장은 본 논문에서 제안하는 test set 생성 기법에 대해 설명한다. 4장에서는 본 연구진의 개발팀이 구현한 OMA DRM System의 시스템 test set생성에 적용한 사례를 통해, 본 기법의 효율성을 보이고자 한다. 마지막으로 5장에서 결론을 맺는다.

2. 관련연구

2.1 Orthogonal Array Testing

Orthogonal Array Testing은 전체 테스트에서 효과적이고 능률적으로 테스트 데이터를 선정하는 방법으로, 시스템적이고 통계적으로 pair-wise한 상호작용으로 테스트한다[2]. 특히 설정 옵션의 조합이 많은 브라우저 등의 시스템의 테스트에 유용하게 사용된다.

기본적으로 기능간의 상호작용과 통합은 defect의 큰 원인이 된다. 이때 발생하는 defect는 대부분 복합적인 상호작용보다 간단한 pair-wise 상호작용으로부터 생기며, 컴포넌트나 설정의 조합들로도 쉽게 놓칠 수 있다. 이러한 이유로, Orthogonal Array Testing을 많이 사용한다. Orthogonal Array Testing은 선택된 모든 변수의 pair-wise한 조합으로 테스트함을 보증하고, 이는 모든 변수의 모든 조합으로 테스트하는 것보다 적은 test case로 효과적이고 간결한 test set을 생성할 수 있다. 이러한 Orthogonal Array Testing은 무엇보다 작성하기 쉬우며, 손으로 작성한 test set보다 오류를 범할 가능성이 낮다.

Orthogonal Array에서 사용하는 용어는 다음과 같이 정의한다[3].

$$L_{Runs}(Levels^{Factors})$$

- Runs: Array의 열의 개수. Orthogonal Array 기법을 통해 생성되는 test case의 개수
- Factors: Array의 행의 개수. Array의 변수의 최대 개수
- Levels: 하나의 Factor가 가질 수 있는 값의 최대 개수

2.2 Equivalence Class Testing

Equivalence Class Testing은 프로그램의 입력 도메인을 equivalence class로 분할하고, 한 클래스 내의 데이터에서 에러가 발생하면 동일 클래스 내의 다른 데이터에서도 에러가 발생하게 될 것이라는 가정하에서 대표되는 test case를 선정하여 테스트를 수행하는 기법이다[3]. Equivalence class는 입력 조건으로 유효하거나 유효하지 않은 집합을 나타낸다.

3. Test Set 생성 기법

Test Set 생성은 다음 그림 1과 같은 절차를 통해 선정된다.

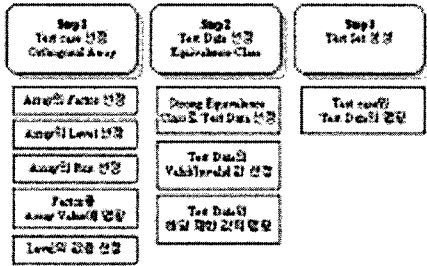


그림 1 Test Set 선정기법

Test Set의 생성은 크게 3단계로 구성된다.

첫 번째 단계는 테스트하고자 하는 시스템의 명세로부터 입력 값으로 지칭할 공통 독립 변수를 선정하여, Orthogonal Array 기법을 이용하여 test case를 선정하는 것이다. 해당 단계에서는 입력 값이 될 독립 변수의 개수를 Factor, 해당 독립변수가 가질 수 있는 값의 최대 개수를 Level로 하여 적용 가능한 Orthogonal Array 중 가장 적은 수를 가지는 Run을 선정하여 Factor와 Level의 값을 Orthogonal Array에 적용시킨다.

두 번째 단계는 첫 번째 단계에서 Orthogonal Array로 구성된 test case에 따라 가질 수 있는 test data를 Equivalence Class Partitioning 기법으로 선정하는 것이다. 첫 번째 단계에서 선정되는 독립변수에 따라 다르게 적용되어야 할 변수들을 고려하여 Equivalence class partitioning을 적용하는 것이다. 본 단계에서 고려되어야 할 요소의 개수를 Cartesian Product를 기반으로 Strong Equivalence Class의 test data를 선정한다. 그리고, 해당 요소들이 가질 수 있는 값의 범위를 통해 유효한 입력 값과 유효하지 않은 입력 값을 분류하여, Strong Equivalence Class로 분류된 test data와 맵핑 시킨다.

마지막 세 번째 단계에서는 Orthogonal Array로 분류된 각 test case에 Equivalence Class로 분류한 test data를 맵핑하여, 완전한 테스트 셋으로 구성한다.

4. 적용사례

4.1 OMA DRM System

DRM (Digital Rights Management) 이란 콘텐츠의 지적 재산권이 디지털 방식에 의해서 안전하게 보호/유지되도록

하여 디지털 콘텐츠가 창작에서부터 소비에까지 이르는 모든 유통 시점에서 거래 규칙과 사용 규칙이 적법하게 성취되도록 하는 기술이다[1].

OMA DRM System은 OMA DRM Spec.을 기반으로 구현된 시스템으로, 미디어 콘텐츠 전송 방식, 전송 매체, 전송과 사용의 동작 및 미디어 콘텐츠의 타입에 따라 다양한 기능을 수행을 수행하고 있어 반복적이고 중복성이 높은 시스템 테스트를 요구한다. 따라서 OMA DRM System의 시스템 테스트에 소요되는 시간 및 비용을 줄이기 위해 Orthogonal Array Testing 기법이 요구 되었으며, 본 논문에서 제안하는 테스트 셋 생성 기법을 적용하기에 적절한 사례가 되었다.

Mobile DRM System에서 미디어 콘텐츠를 다운로드하고 사용하는 단계는 다음 그림 2와 같은 절차를 따른다.

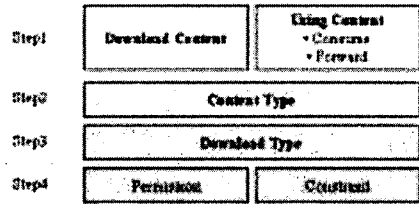


그림 2 OMA DRM System의 사용 절차

그림 2에서 보이는 바와 같이 사용자는 필요한 미디어 콘텐츠를 다운로드 할 것인지 혹은, 이미 다운로드 받은 콘텐츠를 사용할 것인지를 선택하고, 해당 미디어의 타입을 선택한다. 미디어의 다운로드 방식에 따라 사용자는 사용에 제약을 받을 수 있으며, 콘텐츠를 어떻게 얼마나 사용할 것인지를 마지막 단계에서 결정한다.

4.2 Test Set 생성

OMA DRM System 테스트를 위해서는 앞서 설명한 그림 2와 같은 DRM System의 사용 순서에 영향을 받는다.

1) Orthogonal Array 적용

OMA DRM System에서 공통적으로 적용할 수 있는 독립변수와 각 독립변수가 가질 수 있는 값을 OMA DRM spec. 에서 다음 표 1과 같이 추출하였다.

표 1 OMA DRM System의 독립변수

Method	Type	Action	Content
OMA-1 FL	OTA	Download Check	Image
OMA-1 CD	HTTP	Consume	Audio
OMA-2 CD	-	Lost connection	Video
OMA-2 SD	-	Forwarding	Execute

추출한 독립 변수와 해당 값으로 Orthogonal Array 적용을 위한 Factor와 Level 값을 다음과 같이 설정하고, 최소 개수를 가지는 Run을 선정하였다.

$$L_{32}(4^4)$$

선정된 Orthogonal Array에 앞서 선정한 독립변수 및 독립변수 값을 적용한 결과는 다음 표2와 같다.

표 2 Orthogonal Array를 적용한 test case

Method	Run	Result	Comment
OMA 1 R	001	OK	
OMA 1 R	002	OK	
OMA 1 R	003	OK	
OMA 1 R	004	OK	
OMA 1 R	005	OK	
OMA 1 R	006	OK	
OMA 1 R	007	OK	
OMA 1 R	008	OK	
OMA 1 R	009	OK	
OMA 1 R	010	OK	
OMA 1 R	011	OK	
OMA 1 R	012	OK	
OMA 1 R	013	OK	
OMA 1 R	014	OK	
OMA 1 R	015	OK	
OMA 1 R	016	OK	
OMA 1 R	017	OK	
OMA 1 R	018	OK	
OMA 1 R	019	OK	
OMA 1 R	020	OK	
OMA 1 R	021	OK	
OMA 1 R	022	OK	
OMA 1 R	023	OK	
OMA 1 R	024	OK	
OMA 1 R	025	OK	
OMA 1 R	026	OK	
OMA 1 R	027	OK	
OMA 1 R	028	OK	
OMA 1 R	029	OK	
OMA 1 R	030	OK	
OMA 1 R	031	OK	
OMA 1 R	032	OK	
OMA 1 R	033	OK	
OMA 1 R	034	OK	
OMA 1 R	035	OK	
OMA 1 R	036	OK	
OMA 1 R	037	OK	
OMA 1 R	038	OK	
OMA 1 R	039	OK	
OMA 1 R	040	OK	
OMA 1 R	041	OK	
OMA 1 R	042	OK	
OMA 1 R	043	OK	
OMA 1 R	044	OK	
OMA 1 R	045	OK	
OMA 1 R	046	OK	
OMA 1 R	047	OK	
OMA 1 R	048	OK	
OMA 1 R	049	OK	
OMA 1 R	050	OK	
OMA 1 R	051	OK	
OMA 1 R	052	OK	
OMA 1 R	053	OK	
OMA 1 R	054	OK	
OMA 1 R	055	OK	
OMA 1 R	056	OK	
OMA 1 R	057	OK	
OMA 1 R	058	OK	
OMA 1 R	059	OK	
OMA 1 R	060	OK	
OMA 1 R	061	OK	
OMA 1 R	062	OK	
OMA 1 R	063	OK	
OMA 1 R	064	OK	
OMA 1 R	065	OK	
OMA 1 R	066	OK	
OMA 1 R	067	OK	
OMA 1 R	068	OK	
OMA 1 R	069	OK	
OMA 1 R	070	OK	
OMA 1 R	071	OK	
OMA 1 R	072	OK	
OMA 1 R	073	OK	
OMA 1 R	074	OK	
OMA 1 R	075	OK	
OMA 1 R	076	OK	
OMA 1 R	077	OK	
OMA 1 R	078	OK	
OMA 1 R	079	OK	
OMA 1 R	080	OK	
OMA 1 R	081	OK	
OMA 1 R	082	OK	
OMA 1 R	083	OK	
OMA 1 R	084	OK	
OMA 1 R	085	OK	
OMA 1 R	086	OK	
OMA 1 R	087	OK	
OMA 1 R	088	OK	
OMA 1 R	089	OK	
OMA 1 R	090	OK	
OMA 1 R	091	OK	
OMA 1 R	092	OK	
OMA 1 R	093	OK	
OMA 1 R	094	OK	
OMA 1 R	095	OK	
OMA 1 R	096	OK	
OMA 1 R	097	OK	
OMA 1 R	098	OK	
OMA 1 R	099	OK	
OMA 1 R	100	OK	

3) Test Set 생성

OMA DRM System 테스트를 위한 test set의 생성은 앞서 적용한 두 기법으로 얻은 결과들을 맵핑 하여 생성한다. 표2에서 미디어 콘텐츠의 종류에 따라 가능한 permission에 따라 표 3에서 정리한 test data를 맵핑 한다

4.3 분석

4.2절에서 생성한 test set의 효율성을 제안하기 위해 본 절에서는 OMA DRM System 테스트를 위해 개발팀과 독립된 QA(Quality Assurance)조직이 임의적으로 작성한 test set과 본 논문에서 제안한 test set을 이용하여 시스템 테스트를 수행하였다. 시스템 테스트의 결과를 다음 표 4와 같이 얻을 수 있었다.

표 4 OMA DRM System 테스트 결과

	본 논문에서 제안하는 기법	Random 테스트
Test Set 의 개수	64	209
발견한 Defect 수	3	3
보류된 test case 의 수	4	39

2) Equivalence Class 적용

Orthogonal Array를 적용한 test case들은 그림 2의 DRM System 사용 절차에 따라 각각 permission과 그에 해당하는 constraint를 가질 수 있다. Permission과 constraint는 미디어 콘텐츠의 타입에 따라 달리 적용되므로 Equivalence Class를 적용한다. 이들 permission과 constraint를 고려하여 Strong Equivalence Class를 적용하였을 때 얻을 수 있는 test data의 수는 다음과 같다.

- Permission이 가질 수 있는 값: 6
 - Constraint 가 가질 수 있는 값: 7
 따라서, Strong Equivalence Class적용한 test data수: 6 * 7 = 42

Strong Equivalence Class를 적용한 test data에 constraint를 유효한 값과 유효하지 않은 값을 적용하고, Permission의 특성에 따라 적용할 수 없는 constraint를 제외하여, 표 3과 같이 test data를 선정하였다.

표 3 Equivalence Class를 적용한 test data

Run	Permission	Constraint	Value	Result
1	OMA 1 R	OMA 1 R	OMA 1 R	OK
2	OMA 1 R	OMA 1 R	OMA 1 R	OK
3	OMA 1 R	OMA 1 R	OMA 1 R	OK
4	OMA 1 R	OMA 1 R	OMA 1 R	OK
5	OMA 1 R	OMA 1 R	OMA 1 R	OK
6	OMA 1 R	OMA 1 R	OMA 1 R	OK
7	OMA 1 R	OMA 1 R	OMA 1 R	OK
8	OMA 1 R	OMA 1 R	OMA 1 R	OK
9	OMA 1 R	OMA 1 R	OMA 1 R	OK
10	OMA 1 R	OMA 1 R	OMA 1 R	OK
11	OMA 1 R	OMA 1 R	OMA 1 R	OK
12	OMA 1 R	OMA 1 R	OMA 1 R	OK
13	OMA 1 R	OMA 1 R	OMA 1 R	OK
14	OMA 1 R	OMA 1 R	OMA 1 R	OK
15	OMA 1 R	OMA 1 R	OMA 1 R	OK
16	OMA 1 R	OMA 1 R	OMA 1 R	OK
17	OMA 1 R	OMA 1 R	OMA 1 R	OK
18	OMA 1 R	OMA 1 R	OMA 1 R	OK
19	OMA 1 R	OMA 1 R	OMA 1 R	OK
20	OMA 1 R	OMA 1 R	OMA 1 R	OK
21	OMA 1 R	OMA 1 R	OMA 1 R	OK
22	OMA 1 R	OMA 1 R	OMA 1 R	OK
23	OMA 1 R	OMA 1 R	OMA 1 R	OK
24	OMA 1 R	OMA 1 R	OMA 1 R	OK
25	OMA 1 R	OMA 1 R	OMA 1 R	OK
26	OMA 1 R	OMA 1 R	OMA 1 R	OK
27	OMA 1 R	OMA 1 R	OMA 1 R	OK
28	OMA 1 R	OMA 1 R	OMA 1 R	OK
29	OMA 1 R	OMA 1 R	OMA 1 R	OK
30	OMA 1 R	OMA 1 R	OMA 1 R	OK
31	OMA 1 R	OMA 1 R	OMA 1 R	OK
32	OMA 1 R	OMA 1 R	OMA 1 R	OK
33	OMA 1 R	OMA 1 R	OMA 1 R	OK
34	OMA 1 R	OMA 1 R	OMA 1 R	OK
35	OMA 1 R	OMA 1 R	OMA 1 R	OK
36	OMA 1 R	OMA 1 R	OMA 1 R	OK
37	OMA 1 R	OMA 1 R	OMA 1 R	OK
38	OMA 1 R	OMA 1 R	OMA 1 R	OK
39	OMA 1 R	OMA 1 R	OMA 1 R	OK
40	OMA 1 R	OMA 1 R	OMA 1 R	OK
41	OMA 1 R	OMA 1 R	OMA 1 R	OK
42	OMA 1 R	OMA 1 R	OMA 1 R	OK
43	OMA 1 R	OMA 1 R	OMA 1 R	OK
44	OMA 1 R	OMA 1 R	OMA 1 R	OK
45	OMA 1 R	OMA 1 R	OMA 1 R	OK
46	OMA 1 R	OMA 1 R	OMA 1 R	OK
47	OMA 1 R	OMA 1 R	OMA 1 R	OK
48	OMA 1 R	OMA 1 R	OMA 1 R	OK
49	OMA 1 R	OMA 1 R	OMA 1 R	OK
50	OMA 1 R	OMA 1 R	OMA 1 R	OK
51	OMA 1 R	OMA 1 R	OMA 1 R	OK
52	OMA 1 R	OMA 1 R	OMA 1 R	OK
53	OMA 1 R	OMA 1 R	OMA 1 R	OK
54	OMA 1 R	OMA 1 R	OMA 1 R	OK
55	OMA 1 R	OMA 1 R	OMA 1 R	OK
56	OMA 1 R	OMA 1 R	OMA 1 R	OK
57	OMA 1 R	OMA 1 R	OMA 1 R	OK
58	OMA 1 R	OMA 1 R	OMA 1 R	OK
59	OMA 1 R	OMA 1 R	OMA 1 R	OK
60	OMA 1 R	OMA 1 R	OMA 1 R	OK
61	OMA 1 R	OMA 1 R	OMA 1 R	OK
62	OMA 1 R	OMA 1 R	OMA 1 R	OK
63	OMA 1 R	OMA 1 R	OMA 1 R	OK
64	OMA 1 R	OMA 1 R	OMA 1 R	OK

개발된 OMA DRM System에는 총 3가지 타입의 Defect이 존재하였다. Random 테스트는 개발팀에서 오류를 발견할 가능성이 높은 부분에 대해 가능한 한 full coverage를 만족하는 test set을 작성하였다. 표 4에서 보는 바와 같이 본 논문에서 제안하는 기법으로 생성한 test set을 이용하여 실험한 결과는 test set의 수가 Random 테스트의 test set 보다 적고, 모든 종류의 defect을 발견함을 알 수 있다. 또한 보류되는 test case가 적음을 보여줌으로써 본 논문에서 제안하는 기법이 pair-wise한 test case를 생성하여, test case들의 중복성을 줄일 수 있음을 보여준다.

5. 결론 및 향후 연구

본 논문에서는 복잡하고 다양한 설정을 요구하는 시스템 테스트를 위해 Orthogonal Array와 Equivalence Class 기법을 이용하여, 효율적인 test set을 생성하는 방법을 제안하였다. 그리고, 제안한 기법을 OMA DRM System에 적용한 사례를 통해 제안한 기법이 test set의 중복성을 줄이며, 효율적으로 시스템에 존재하는 defect을 찾을 수 있음을 보였다. 추후 본 논문의 기법을 잘 정비하여, 타 시스템의 테스트에 적용할 수 있도록 하겠다.

6. 참고문헌

[1] OMA-Download-DRM-v2.0, Open Mobile Alliance™, <http://www.openmobilealliance.org>, 20 Apr.2004
 [2] Orthogonal Array Testing Strategy Technique, Jeremy M.Harrell, <http://www.seilevel.com/OATS.html>
 [3] A Library of Orthogonal Arrays, Sloane, Neil J.A., Information Sciences Research Center, AT&T Shannon Labs. 9 Aug. 2001, <http://www.research.att.com/~njas/oadir/>
 [4] Software Testing and Continuous Quality Improvement, William E. Lewis, Boca Raton London New York 2000.