

추상 테스트 케이스 성숙도 모델 기반의 테스트 케이스 추적성 연구

박지훈^o, 손현승, 김영철
홍익대학교 소프트웨어공학연구소
pjh@selab.hongik.ac.kr, hson@live.co.kr, bob@hongik.ac.kr

A Study on the Test Case Traceability based on Abstract Test Case Maturity Model

Jihoon Park^o, Hyun-Seung Son, R. Young Chul Kim
Software Engineering Lab., Hongik University

요 약

최근 고품질을 위한 요구사항의 추적성이 대두 되는 실정이다. 요구사항 기반인 테스트 케이스로 고객의 원하는 올바른 소프트웨어(the right system)인지를 검증 한다. 그러나 아직도 SI 중소/벤처/스타트업 기업들은 풍부한 테스트 케이스 확보의 필요성이 매우 크다. 이 해결을 위해, 유스 케이스 모델 기반의 상위 및 하위 모델을 적용한 테스트 케이스를 식별한다. 하지만 빈번한 요구사항 수정은 모델들 변경 및 테스트 케이스 변경이 요구 된다. 즉 요구 사항 기반의 모델 메커니즘 기반으로 테스트 케이스를 추출하는 방법을 제안한다. 이를 통해 상위 레벨부터 하위 레벨의 테스트 케이스 발생 메커니즘과 그에 따른 추적성을 확보할 수 있다.

1. 서 론

소프트웨어 시스템의 신뢰성과 안전성을 위해서는 시스템이 오류를 일으킬 수 있는 결함을 가능한 많이 찾아내어 수정해야 한다. 최종 제품의 품질과 소프트웨어를 효과적으로 재사용하기 위해서는 철저한 테스트를 통한 검증이 되어야 한다. 소프트웨어의 복잡도가 증가하면서 수많은 기능들이 소프트웨어에 포함되고 서로 상호작용하면서 결함이 증가하고 있다. 결함이 증가하면서 개발 기간과 유지비용이 증가하고 있다. 높은 신뢰도를 필요로 하는 항공이나 의료 시스템의 경우 테스트 단계에서 전체 프로젝트 개발 자원의 50~70%를 차지하기도 한다[1]. 이처럼 테스트 전략과 관계없이 대형 시스템은 본질적으로 테스트가 복잡하다. 테스트 케이스를 작성하는 것은 테스트에서 가장 어려운 단계일 수 있다. 더 나은 품질의 소프트웨어를 얻기 위해서는 테스트 비용과 노력을 줄이고 많은 수의 테스트 케이스를 설계하고 테스트를 수행해야 한다[2]. 그로 인해 소프트웨어 테스트의 중요성이 부각되고 있으며 최근 테스트 분야는 신종 산업으로 급성장하고 있다.

국내 SI 중소/벤처/스타트업 기업들은 여전히 풍부한 테스트 케이스 확보의 필요성을 요구한다. 전통적으로 소프트웨어 테스트는 프로그램 소스코드나 자연어를 기반으로 하는 매우 추상화된 명세를 통해 이루어진다. 이상적인 테스트는 코드가 아니라 요구사항이나 명세서 문서를 가지고 최종 산출물의 결과와 일치 혹은 불일치 여부를 보여주는 것을 목적으로 해야 한다[3]. 하지만 요구

사항은 다양한 외부적 요인으로 사용자들로부터 잦은 변경이 일어난다. 이를 해결하기 위해 유스 케이스 모델 기반의 상위 및 하위 모델을 적용하여 테스트 케이스를 식별한다. 하지만 잦은 요구사항의 수정은 모델 및 테스트 케이스의 잦은 변경을 야기한다. 효율적인 테스트를 위해서는 소프트웨어 개발 프로세스 간의 추적성을 보장해야 한다[4]. 따라서 요구사항을 기반으로 생성된 모델로부터 테스트 케이스를 생성하여 시스템이 요구사항에 적합하게 작동하는지를 검증하는 모델 기반 테스트가 필요하다. 본 논문에서는 모델 메커니즘을 기반으로 상위 레벨부터 하위레벨의 테스트 케이스 추적성을 보장하는 추상 테스트 케이스 성숙도 모델[8]을 추출하는 방법을 제안한다. 요구사항 명세서에서부터 유스케이스 다이어그램, 시퀀스 다이어그램, 액티비티 다이어그램을 순차적으로 추출할 수 있다. 액티비티 다이어그램에서 추출한 액티비티 경로들로 테스트 시나리오를 생성할 수 있다. 각 테스트 시나리오에서는 메서드의 테스트 유닛을 발생할 수 있다. 테스트 유닛에는 선 상태와 후 상태의 입력을 고려하여 다양한 테스트 케이스 생성이 가능하다. 이를 통하여 테스트 커버리지 향상과 잦은 요구사항 변경에도 테스트 케이스 추적성을 통하여 테스트 케이스 생성 및 수정이 가능하다.

본 논문은 총 4장으로 이루어져 있다. 2장은 테스트 케이스를 생성하기 위한 관련 논문들에 대한 내용이다. 3장은 요구사항에서 테스트 케이스까지의 모델 메커니즘을 적용하는 방법에 대한 내용을 설명한다. 4장에서는 결론 및 향후연구이다.

2. 관련연구

M. Sarma et al.의 논문[5]에서는 상태 다이어그램 형태의 SDG(Sequence Diagram Graph)를 정의하여 시퀀스 다이어그램의 시나리오를 표현한다. SDG는 S, Σ, q0, F로 이루어져 있다. S는 운영 시나리오의 다양한 상태를 나타내는 모든 노드의 집합이다. Σ는 하나의 상태에서 다른 상태로의 전이를 나타내는 엣지의 집합이다. q0는 작업이 시작되는 상태를 나타내는 초기 노드이다. F는 작업이 종료되는 상태를 나타내는 최종 노드의 집합이다. 시퀀스 다이어그램에서 추출된 SDG에 대한 테스트 알고리즘을 적용하여 테스트 케이스를 도출한다. 하지만 시퀀스 다이어그램의 다양한 결합 조각을 고려하지 않고 메시지만을 이용하여 조건을 결정하는 단점이 있다.

J. Hartmann et al.의 논문[6]에서는 UML(Unified Modeling Language)의 상태와 같이 완성된 소프트웨어의 실행모델로부터 테스트 케이스를 도출한다. 그러므로 사용자가 요구했던 요구사항과 테스트 케이스가 다를 수 있다. 개발에 참여하지 않은 테스터의 경우 규모가 큰 소프트웨어의 복잡한 다이어그램만으로 테스트 케이스를 도출해내야 한다.

안성빈의 논문[7]에서는 설계 모델이 테스트 설계에 사용되도록 테스트 유닛을 정의했다. 테스트 유닛은 Design Unit을 재정의한 것으로 시퀀스 다이어그램에서 추출할 수 있다. 그 중 Method's Test Unit은 객체에서 발생시키는 간단한 메시지로 테스트 케이스 템플릿을 통해 테스트 케이스 생성이 가능하다. 하지만 최소의 테스트 케이스로 테스트를 진행하기 때문에 테스트 커버리지나 예상치 못한 결함에 취약하다.

본 논문에서는 그림 1과 같이 요구사항 명세서부터 UML Diagrams를 통하여 테스트 케이스를 생성하는 방법을 제안한다. 모델 메커니즘을 적용하여 [5]의 메커니즘에 [7]의 메커니즘을 접목하여 요구사항부터 테스트 케이스의 추적성을 확보하여 [6]의 단점을 보완하려 한다.

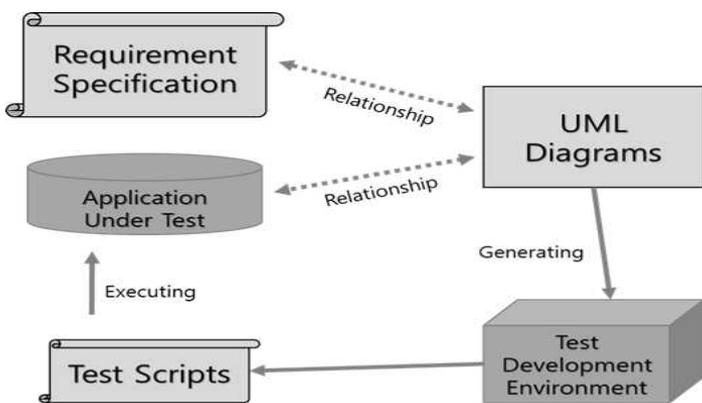


그림 1 UML Diagram을 통한 요구사항과 테스트의 관계

3. 모델기반의 테스트 케이스 추적성 매트릭스

모델기반의 테스트 케이스 추적성 매트릭스는 요구사항으로부터 테스트 케이스까지 모델 변환을 통해 추적성을 확보하는 방법이다. 그림 2와 같은 모델 변환 프로세스를 제안한다.

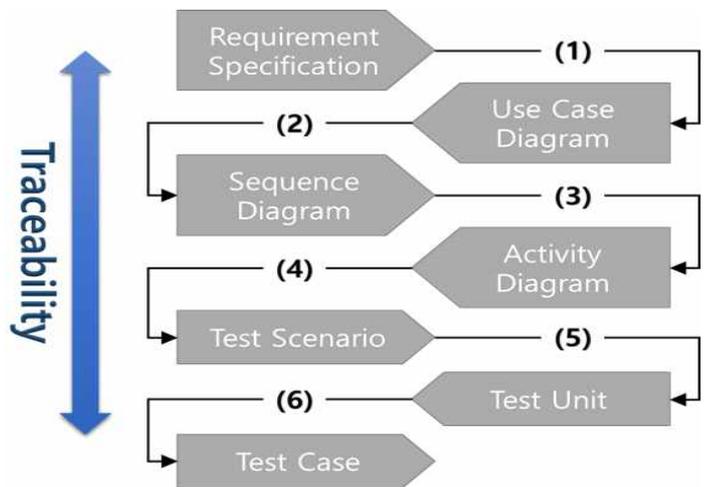


그림 2 테스트 케이스 추적성을 위한 모델 변환 프로세스

- (1) 요구사항 명세서는 시스템이 어떤 속성이나 특성을 가지고 동작해야 하는지에 대한 설명이다. 이는 유스 케이스 다이어그램으로 변환될 수 있다. 시스템을 이용하는 액터와 처리기능인 유스 케이스들 사이의 관계를 나타낸다. 이를 위해서는 목표 지향 요구사항의 형식이 필요하다[7].
- (2) 유스 케이스 다이어그램은 시퀀스 다이어그램으로 변환될 수 있다. 유스 케이스에 대한 이벤트 시퀀스를 정의한다. 각 유스 케이스마다 시퀀스 다이어그램이 생성된다. 시퀀스 다이어그램의 시스템 객체들 간 어떤 메시지가 어떤 순서로 발생하는지 나타낸다.
- (3) 시퀀스 다이어그램은 액티비티 다이어그램으로 변환될 수 있다. 시퀀스 다이어그램은 객체 간 메시지들의 집합이다. 시퀀스 다이어그램의 Message와 각 메시지의 Send/Recieve를 액티비티 다이어그램의 Node와 Control flow로 변환한다. loop, alt, opt들은 Decision node로 변환한다.
- (4) 액티비티 다이어그램은 테스트 시나리오로 변환될 수 있다. 액티비티 다이어그램에서 추출된 액티비티 경로는 input에서부터 output까지의 모든 경로이다. 액티비티 경로를 모두 체크하면 테스트 커버리지를 만족할 수 있는 테스트 시나리오를 생성할 수 있다.
- (5) 테스트 시나리오는 테스트 유닛들로 변환될 수 있다. 테스트 시나리오는 액티비티 다이어그램 Node들의 흐름이다. 액티비티 다이어그램의 Node들은 시퀀스 다이어그램에서의 Message이다. 이는 각 객체의 Method를 의미하므로 테스트 유닛[7]으로 볼 수 있다.
- (6) 테스트 유닛은 테스트 케이스로 변환될 수 있다. 각 테스트 유닛들은 Carlson이 정의한 Design Unit을 재정의한 것이다. 테스트 유닛에는 Method, Reusable Pattern, State, Maximum Linear Unit, Dialogue로 이루어져 있는데 이 매트릭스에서는 method에 대한 테스트 유닛 정의를 사용한다[7].

그림 3은 요구사항부터 모델 메커니즘을 적용하여 테스트 케이스까지의 추적성을 보여주는 예시이다. 각각의 모델 요소가 다음 모델의 어느 부분으로 변환되는지 보여준다.

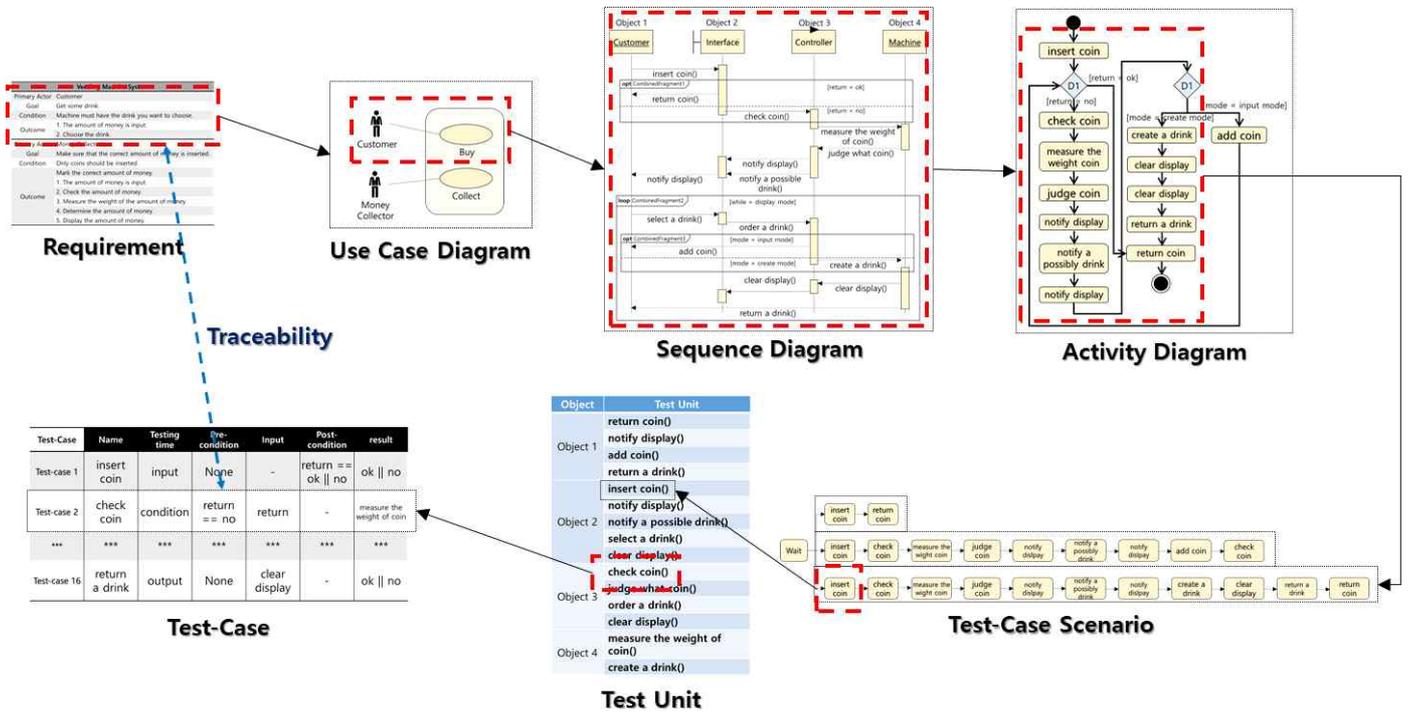


그림 3 모델 기반의 상위 레벨부터 하위 레벨 테스트 케이스의 추적성

4. 결론 및 향후 연구

참 고 문 헌

시스템 구조가 복잡해짐에 따라 테스트의 범위는 더 광범위해지고 있다. 또한 잦은 요구사항의 변경으로 인해 테스트 비용과 시간을 증가시켜 효율적인 테스트가 쉽지 않다. 이를 해결하기 위해서는 테스트 케이스 추적성을 통한 관리가 필요하다.

본 논문에서는 모델 기반의 요구사항으로부터 테스트 케이스까지의 추적성 매트릭스를 제안한다. 개발 중 잦은 요구사항의 변경에도 추적성을 이용하여 테스트 케이스의 생성 및 삭제가 수월해진다. 추적성 매트릭스를 총 6개의 프로세스로 세분화하였다. 요구사항을 통해 세부적인 모델링을 추출하여 정의한 테스트 유닛을 통한 테스트 케이스를 추출한다. 과거에는 최소의 테스트 케이스를 이용하여 테스트 커버리지를 충족시키는 방안이었다면 이제는 모든 테스트 케이스를 추출하여 모든 커버리지를 충족시키는 방안을 제시하고자 한다.

향후 연구에서는 다양한 테스트 커버리지를 적용할 수 있는 모델 변환 규칙을 정제하여 자동화 할 수 있는 도구를 구현할 것이다. 또한 Goal Oriented Requirement Engineering을 통하여 좀 더 명확한 요구사항부터 테스트 케이스까지의 추적성 매트릭스를 연구할 것이다.

ACKNOWLEDGEMENT

이 논문은 2015년 교육부와 한국연구재단의 지역혁신창의인력양성사업(NRF-2015H1C1A1035548)과 2017년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2017R1D1A3B03035421).

[1] S. Rayadurgan and Mats. P. E. Heimdahl, "Test-Sequence Generation from Formal Requirement Models", Proceedings of the 6th IEEE International Symposium on High Assurance Systems Engineering(HASE' 01), pp.23-31, October, 2001.

[2] H. Shokry and M. Hinchey, "Model-based Verification of Embedded Software", IEEE Computer Society on the Irish Software Engineering Research Centre, pp.53-59, April, 2009.

[3] S. Jung, H. Noh and C. Yoo, "Test Case Generation Technique Based on SysML State Machine Diagram", The Korean Institute of Information Technology 9th Journal No 3, pp.31-39, March, 2011.

[4] 박보경, 권하은, 문소영, 이유진, 김영수, 이상은, 박용범, 김영철, "요구사항 추적성을 위한 요구사항 추적 모델", 한국정보처리학회, 2015

[5] M. Sarma, D. Kundu and R. Mall, "Automatic Test Case Generation from UML Sequence Diagrams", Proc. of the 15th International Conference on Advanced Computing and Communications, pp.60-67, December, 2007

[6] J. Hartmann, M. Vieira, H. Foster and A. Ruder "A UML-based Approach to System Testing", Innovation in Systems and Software Engineering Springer Volume 1, pp12-24, April, 2005

[7] 안성빈, "Requirement Based Testing On Use Case Paradigm", 홍익대학교 일반대학원 석사학위논문, 2011

[8] 박보경, 장우성, 김기두, 이근상, 김영철, C. R. Carlson, "한국형 테스트 모델을 위한 경량화 연구", KCSE 2017 vol 19, pp318-319, February, 2017



2017 Korea Software Congress

Date

2017. 12. 20(수)~22(금)

Venue

부산 벅스코 컨벤션홀

Theme

소프트웨어, 4차 산업혁명의 열쇠



주요행사

12.20

Keynote-Editor in Chief Hamido Fujita(Elsevier)
 Invited Talk-허재혁 교수(KAIST)
 배현섭 사장(슈어소프트테크)
 김상현 부사장(한국오라클)
 특별세션-신진교수 최선연구소개
 데이터베이스 및 데이터 마이닝 최신기술 워크샵
 지능형 사물인터넷 기반 융합 서비스 기술 동향
 머신러닝연구회 동계 워크샵
 SW+City: Smart City Reloaded
 국제표준 기반 오픈 데이터 유통 플랫폼 확장 기술 개발 워크샵
 우수 국제학술회의 목록 갱신 토론회

12.21

Keynote-손상혁 교수(DGIST)
 President Hironori Kasahara(2018 IEEE Computer Society)
 Invited Talk-하정우 리더(네이버)
 전진욱 사장(비트컴퓨터)
 김태훈 본부장(KIAPS)
 김윤재 과장(기상청)
 Dr. Kohtaro Asai(Mitsubishi Electric Corporation)
 특별세션-신진교수 최선연구소개
 SW 구현/데모 경진대회(대학원생)
 군집지능(SAAL)플랫폼 요구 분석 워크샵
 인공지능 국가전략프로젝트 개방형 평가
 비디오튜링테스트(VTT)를 위한 비디오 이해 답러닝기술 워크샵

12.22

SW 구현/데모 경진대회(학부생)
 학부생주니어논문경진대회

Sponsored by



| 세션 | 시간 | 분야 | 장소 | 좌장 |
|----------|---|-----------|------|---------|
| 22B3B4 | 14:00-17:30 | 소프트웨어공학II | 103호 | 홍신(한동대) |
| 22B3B4-1 | 사물 인터넷 기기의 애플리케이션 설계 및 개발을 위한 패러다임의 정량적/정성적 분석 및 비교 안윤근 · 이육진(한양대) | | | |
| 22B3B4-2 | 자동 소프트웨어 프로세스 구축을 위한 공개 소프트웨어 툴 체인 연구 이진협 · 장우성 · 김영철(홍익대) | | | |
| 22B3B4-3 | 모바일 클라이언트 통합테스트를 위한 애스펙트 구성 방법 최윤석(동국대), 최은만(동국대) | | | |
| 22B3B4-4 | 임베디드 소프트웨어를 위한 모듈의 분리성 메트릭 정병훈, 김정선(한양대) | | | |
| 22B3B4-5 | 추상 테스트 케이스 성숙도 모델 기반의 테스트 케이스 추적성 연구 박지훈 · 손현승 · 김영철(홍익대) | | | |

| 세션 | 시간 | 분야 | 장소 | 좌장 |
|----------|--|--------|------|-------------------|
| 22C1C2 | 09:00-12:30 | 언어공학II | 104호 | 김유섭(한림대) 이상곤(전주대) |
| 22C1C2-1 | 셀프 매칭 어텐션 기반 포인터 네트워크를 이용한 한국어 상호참조해결 박천음(강원대), 김현기(ETRI) 이창기(강원대) | | | |
| 22C1C2-2 | 대화 단위의 화행 분석을 위한 RNN-CNN 기반 한국어 화행 분석 시스템 윤정민(동아대), 고영중(동아대) | | | |
| 22C1C2-3 | 기분식 어절사전 축소를 통한 한국어 형태소 분석에서의 메모리 및 수행 시간 최적화 김해민 · 양선 · 고영중(동아대) | | | |
| 22C1C2-4 | S ³ -Net: SRU 기반 문장 및 셀프 매칭 네트워크를 이용한 한국어 기계독해 박천음(강원대), 배경훈·장재용·홍윤기(LGU+), 홍수린·황이규(마인즈랩), 김현기(ETRI), 이창기(강원대) | | | |
| 22C1C2-5 | Word2Vec를 이용한 주제별 감성어 사전 구축 장환석 · 정광용 · 장은영((주)다음소프트) | | | |
| 22C1C2-6 | Action 기반 Local attention mechanism을 이용한 linear time 한국어 형태소 분석 황현선 · 이창기(강원대) | | | |
| 22C1C2-7 | Probase를 활용한 OpenDirectoryProject 기반의 텍스트 분류 전소영 · 이지민 · Aliyeva Dinara · 이상근(고려대) | | | |
| 22C1C2-8 | Word2Vec를 이용한 감성어 분석 방법 장환석 · 장은영 · 정광용 ((주)다음소프트) | | | |
| 22C1C2-9 | 질문 분석을 위한 위키피디아 기반 카테고리 별 속성 자원 구축 공태용, 김민지, 류범모 | | | |

| 세션 | 시간 | 분야 | 장소 | 좌장 |
|----------|--|----------|------|----------------------|
| 22C3C4 | 14:00-17:30 | 사물인터넷III | 104호 | 서동만(대구가톨릭대) 박유현(동의대) |
| 22C3C4-2 | Energy Efficient Crowdsourcing for Internet of Things Applications over NB-IoT Networks SarderFakhrulAbedin.Md.ShirajumMunir. Md. Golam Rabiul Alam . Choong Seon Hong (Kyung Hee University) | | | |
| 22C3C4-3 | RNN based Energy Demand Prediction for Smart-Home in Smart-Grid Framework Md. Shirajum Munir · Sarder Fakhrul Abedin · Md. Golam Rabiul Alam · Do Hyeon Kim · Choong Seon Hong | | | |
| 22C3C4-4 | 이기종IoT플랫폼간상호운용성을위한 oneM2M 아키텍처 기반의 연동 시스템 강성수(광운대), 박지우(광운대), 정광수(광운대) | | | |
| 22C3C4-5 | IoT 환경에서 oneM2M 플랫폼을 이용한 웹 기반 리소스 모니터링 시스템 이만성(광운대), 김건우(광운대), 박지우(광운대), 정광수(광운대) | | | |
| 22C3C4-6 | 고효율 고신뢰 사물인터넷 통신을 위한 IETF 최신 연구 동향 분석 및 적용 김서향, 박준현, 변형호, 오영준, 김종권 (서울대) | | | |
| 22C3C4-7 | XACML 정책 작성시 요청에 따른 효과적인 정책 평가 요인 수집 방법 오용택(한양대학교), 이육진(한양대학교) | | | |