Fall Conference 2012
제38회 한국정보처리학회
추계학술발표대회
논문집(하)

일자: 2012년 11월 22일(목)-23일(금)
장소: 제주대학교 아라캠퍼스
주최: 사단법인 한국정보처리학회
주관: 
후원: nipa

협찬: 롯데정보통신, 삼성SDS, SK텔레콤, 곳모닝아이테크, 곳센터코칼라지, 낙스테크, 비트컴퓨터, 영우디지털, 유비벨록스, 아나텍, 제이컴정보, 콤텍시스템, 테크그룹, 티맥스소프트, 한국IT감리컨설팅, 한글크컴퍼터, 효성인포메이션시스템, NHN, KCC정보통신
<table>
<thead>
<tr>
<th>제목</th>
<th>저 자</th>
<th>연도</th>
</tr>
</thead>
<tbody>
<tr>
<td>정보처리정보학(IT 교육, 등)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>관련연구 분석을 통한 e-러닝 교육 시스템 설계</td>
<td>KIPS_C2012J_0001</td>
<td>1583</td>
</tr>
<tr>
<td>군집화를 이용한 하이브리드 기반 체계설계 방법 기법</td>
<td>KIPS_C2012J_0002</td>
<td>1587</td>
</tr>
<tr>
<td>소셜 커뮤니티 로봇 관련기의 설계 및 구현</td>
<td>KIPS_C2012J_0005</td>
<td>1591</td>
</tr>
<tr>
<td>물류정보통합기반의 Smart SCM 모델에 관한 연구</td>
<td>KIPS_C2012J_0022</td>
<td>1595</td>
</tr>
<tr>
<td>디스크 I/O 성능에 따른 가상 서버 통합에 대한 고찰</td>
<td>KIPS_C2012J_0042</td>
<td>1599</td>
</tr>
<tr>
<td>PRO 시스템의 설계 및 구현</td>
<td>KIPS_C2012J_0059</td>
<td>1603</td>
</tr>
<tr>
<td>IMO 회원국들 대상으로 한 효율적인 해운 물류 및 보안 강화를 위한 RFID 기반 Ubiquitous Port 적용 사례 연구</td>
<td>KIPS_C2012J_0081</td>
<td>1607</td>
</tr>
<tr>
<td>스마트폰 기반의 모바일 매출업 개발 등</td>
<td>KIPS_C2012J_0093</td>
<td>1611</td>
</tr>
<tr>
<td>컴퓨터공학기술 분야 종합설계 교과목 총괄생 영역 평가 방안</td>
<td>KIPS_C2012J_0103</td>
<td>1615</td>
</tr>
<tr>
<td>SNS환경에서 CRM 마케팅 활성화를 위한 요인 분석</td>
<td>KIPS_C2012J_0116</td>
<td></td>
</tr>
<tr>
<td>PDCA 모델과 SMART 조건 기반의 자기경영 시스템 설계 및 구현</td>
<td>KIPS_C2012J_0137</td>
<td></td>
</tr>
<tr>
<td>활동 데이터 수집을 통한 한소비출량 신정방법 및 구현</td>
<td>KIPS_C2012J_0151</td>
<td></td>
</tr>
<tr>
<td>KSCD를 활용한 KAIST 연구 활동 분석</td>
<td>KIPS_C2012J_0169</td>
<td></td>
</tr>
<tr>
<td>국내 논문 저자의 소속 연구기관 분석을 위한 시스템</td>
<td>KIPS_C2012J_0172</td>
<td></td>
</tr>
<tr>
<td>SNS기반 e-Learning시스템 설계 및 개발</td>
<td>KIPS_C2012J_0194</td>
<td></td>
</tr>
</tbody>
</table>
UML2.4.1 기반 메시지-순차적 다이어그램을 통한 테스트 케이스 추출 연구

우수정*, 김동호, 손현승, 김영철
*홍익대학교 일반대학원 소프트웨어공학연구실
 e-mail : {woo, bob}@selab.hongik.ac.kr

Text Case Extraction with Message Sequence Diagram (MSD)
based on UML2.4.1

SuJeong Woo*, D. H. Kim, S. H. Son, Robert Young Chul Kim
*Dept. of Computer Information & Comm., Hongik University

요 약
기존 연구에서는 순차적, 상태, 엑티브 다이어그램 기반의 테스트 케이스 추출을 초점으로 두고 있다. 하지만 현재 회사의 모델링 언어인 UML2.4.1(Unified Modeling Language) 기반으로 한 테스트 케이스 추출 메커니즘은 없다. 그래서 본 논문은 UML2.4.1 기반의 기존의 본인-결과 다이어그램의 접근을 통해 테스트케이스 추출 메커니즘을 제안한다. 이를 위해 UML2.4.1의 메시지-순차적 다이어그램에 ECA Rule(Event Condition Action) 기법을 적용하고, 제한한 집합 고유리즘을 통해 확장된 메시지-순차적 다이어그램을 위안-결과 다이어그램과 접목한 후, 결정 테이블로 테스트케이스를 발생한다. 이러한 결과물로 테스트케이스 생성기에서 테스트케이스 추출과 자동화 와이어 시스템을 적용한다.

I. 서론
최근 앱에서 소프트웨어는 우리의 생활에 있어 스마트 홈, 스마트 TV, 스마트폰, 셀프 맥스 등과 같은 여러 분야에 걸쳐 적용되고 있다. 그 중에서도 가장 많은 사용률은 앱에서 소프트웨어 분야에 적용한 사례가 증가하고 있다. 그 적용한 사례는 현대 IBM에서 GM의 신형 전기차 '에보레 볼트' 개발에 직접 참여하여 1000만 대의 소프트웨어로 들어가는 앱에서 소프트웨어를 개발하고 있다[6].

이와 같이 실시간, 복잡성, 확장성, 자동차 앱에서 앱을 사용되어 많은의 앱을 사용하고 있는 Unified Modeling Language 2.4.1(UML2.4.1)을 사용하여 테스트케이스를 수행하고자 한다. UML2.4.1의 여러 언어 중 본 논문은 메시지-순차적 다이어그램을 통하지 테스트케이스를 추출한다. UML2.4.1의 메시지-순차적 다이어그램은 여러 객체 사이에 메시지를 보내는 순서를 명확하게 했으며 프레임isphere를 적용시켜 다이어그램의 변화를 위한 지정된 정소를 제공한다[2]. 하지만 메시지-순차적 다이어그램을 통한 테스트케이스 추출은 불가능하기 때문에 메시지-순차적 다이어그램을 ECA Rule(Event Condition Action)의 기법을 적용하여 확장한다. ECA Rule 기법은 메시지-순차적 다이어그램의 각 객체에 Pre-Condition, Action, Post-Condition로 세부화 하는 기법이다[10]. 확장된 메시지-순차적 다이어그램을 테스트케이스 발생이 가능한 위안-결과 다이어그램에 접목 시킨다[4]. 위안-결과 다이어그램의 접목을 통해 발생시키는 것은 최소 제공 테스트케이스로 100%의 기능적인 요구사항 채용지를 만족 시킬 수 있는 지정된 방법[3].

본 논문의 구성은 다음과 같다. 2장은 테스트케이스 추출 과정에 3장은 사례연구, 마지막 4장은 결론 및 향후 연구에 대해서 기술한다.

2. 테스트케이스 추출 과정
본 장에서는 테스트케이스 추출 과정에 대해 설명한다.

그림 1은 테스트케이스 추출 과정이다. 먼저 신속 단계는 메시지 순차적 다이어그램을 ECA Rule을 적용하여 확장한다. 두 번째 단계는 확장된 메시지 순차적 다이어그램을 위안-결과 다이어그램을 접목하여

* 본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT 연구센터 지원사업(NIPA-2012-(H0301-12-3004))과 교육과학기술부와 한국연구재단의 지역혁신인력양성사업으로 수행된 연구결과로부터.
테스트케이스 발생시킨다.

(그림 1) 테스트케이스 추출 과정

다음은 각 순서에 대한 설명이다.

1 단계: ECA Rule 을 적용한 메시지-순차적 다이어그램 작성.

(그림 2) 1:1 메시지-순차적 다이어그램


그림 3은 Alternative의 확장된 메시지-순차적 다이어그램이다. if/else 를 나타내기 위해 A 객체 안에 Post Condition 1, 2로 구분한다. 또한 B 객체에서 나가는 Action 메시지 부분은 "<<"로 한다.

(그림 3) Alternative 메시지-순차적 다이어그램

그림 4는 Option의 확장된 메시지-순차적 다이어그램이다. if만 나타내는 그림 3과 비슷한 메시지-순차적 다이어그램이다. A 객체 안에 Post Condition 1, 2로 구분하고, B 객체에서 나가는 Action 메시지 부분은 ">>"로 한다.

(그림 4) Option 메시지-순차적 다이어그램

2 단계: 확장된 메시지-순차적 다이어그램과 원인-결과 다이어그램의 접목 및 테스트케이스 발생


(그림 7) 메시지-순차적 다이어그램의 요소

그림 7과 8은 메시지-순차적 다이어그램과 원인-결과 다이어그램의 요소를 나타낸 것이다. 요소들을 통해서 나타낸 알고리즘은 다음과 같다.

(그림 8) 원인-결과 다이어그램 요소

표 1은 메시지-순차적 다이어그램과 원인-결과 다이어그램의 접목 알고리즘이다. 메시지-순차적 다이어그램의 Event 메시지를 잃는다. 만약 메시지-순차적 다이어그램의 Return 메시지가 아무것도 없을 경우 종료된다. 아니면 다음과 같이 수행된다. 이때 메시지-순차적 다이어그램의 Return 메시지가 끝나지 않을 때까지 수행한다. 만약 메시지-순차적 다이어그램의 ECA Rule의 Action 부분이 Alternative이면 원인-결과 다이어그램의 InputMessage는 "Evetn"이란 하여, Condition은 "alt"이라고 표시한다. 그리고 메시지-순차적 다이어그램의 Lifeline의 Action을 수행한 후 원인-결과 다이어그램의 OutMessage은 메시지-순차적 다이어그램의 Return 메시지를 표시하는데 alt의 경우는 두가지의 메시지를 나타내게 된다.
재38회 한국정보처리학회 추계학술발표대회 논문집 제19권 2호 (2012. 11)

途中 함수 알고리즘

//MSD : Message Sequence Diagram
//CE : Cause-Effect Diagram

Read MSD.Message.Event;
if(MSD.Message.Return == Null){
    exit(0);
}
else{
    while(MSD.Message.Return != Null){
        if(MSD.Lifeline.ECARule.Message.Action == alt)
            {CE.InputMessage == "Event";
             CE.Condition = "alt";
             MSD.Lifeline.Action();
             CE.OutputMessage1 <- MSD.Message.Return1;
             CE.OutputMessage2 <- MSD.Message.Return2;
            }
        else if(MSD.Lifeline.ECA.Rule.Message.Action == alt)
            {CE.InputMessage == "Event";
             CE.Condition = "opt";
             MSD.Lifeline.Action();
             CE.OutputMessage1 <- MSD.Message.Return1;
            }
    }
}

그림 10은 Alternative 접목에 대해 나타낸 것이다. 사용자는 드라이브(D) 비트 또는 리턴(R) 비트를 눌러 이 방식 '앞으로' 가거나 '뒤로' 가게 하는 것이다. 이 때 인원·결과 다이어그램의 Condition 부분은 'a'를 표시하여 alt를 나타낸다.

그림 9는 메시지-순차적 다이어그램에 대한 테스트케이스 발생에 대한 것이다.

그림 12는 Option 접목에 대해 나타낸 것이다. 연료가 10 미만이 있을 경우 표시하여 사용자가 볼 수 있도록 한다. 이때 인원-결과 다이어그램은 두 가지 조건에 맞게 Output을 나타낸다. opt 접목이기 때문에 'o'표시를 해둔다.

표 2는 1:1 접목의 테스트케이스다. 총 2 가지의 테스트케이스가 발생한다.
3. 사례연구

복잡한 메시지-순차적 다이어그램의 테스트케이스 발생 사례연구로서 그림 14 와 같이 자동차 와이파이에 대한 테스트케이스 추출에 대해 연구한다. 비가 오 경우 사용자가 왜이파 비트를 놓였을 때 센서가 작동 된다. 센서는 비가 많이 오거나 적게 오거나 토마한 후, 왜이파를 빨리 움직이거나, 전하히 움직이는 경우를 나타낸 것이다. 이때의 테스트케이스는 표 5 와 같이 발생 한다. 총 12 가지의 테스트케이스가 발생되며 Option 에 관하여 4 가지 Alternative 에 관하여 8 가지이다.

(그림 14) 복잡한 메시지-순차적 다이어그램과 원인-결과 다이어그램 볼록 

<표 5> 복잡한 접목의 테스트케이스

<table>
<thead>
<tr>
<th>NO</th>
<th>Post condition</th>
<th>Test Condition</th>
<th>Expectation Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC1</td>
<td>11-Fuse=10 +F</td>
<td>opt</td>
<td>O1-Sign=F</td>
</tr>
<tr>
<td>TC2</td>
<td>11-Fuse=10 +T</td>
<td>opt</td>
<td>O1-Sign=T</td>
</tr>
</tbody>
</table>

적용하여 확장하였다. ECA Rule 기법의 Pre-Condition, Action, Post-Condition 을 메시지-순차적 다이어그램 2.4.1 에 적용시켜 확장시킨다. 확장된 메시지-순차적 다이어그램은 테스트케이스가 발생 가능한 원인-결과 다이어그램과 접목시켰다. 접목한 원인-결과 다이어그램을 통해서 테스트케이스 추출하였다.

함께 연구를 확장된 메시지-순차적 다이어그램과 원인-결과 다이어그램과 메타모델정의를 한다. 메타모델을 통해 알고리즘을 구현할 것이다.

참고문헌


4. 결론 및 향후 연구

UML2.4.1 버전의 메시지-순차적 다이어그램은 여러 객체 사이에 메시지 보내는 순서를 명확하게 했으며 프레임 엘리먼트를 적용시키기 다이어그램의 레이어를 위한 지속된 장소를 제공한다[2]. 총 11 가지의 컨버전드 프레임을 통해 간결하고 여러 객체 사이에 메시지 보내는 순서가 명확하게 가능하다[2].

본 논문의 메시지-순차적 다이어그램 2.4.1 을 통해 서 테스트케이스 발행시키기 위해 ECA Rule 기법을 적용하여 확장하였다. ECA Rule 기법의 Pre-Condition, Action, Post-Condition 을 메시지-순차적 다이어그램 2.4.1 에 적용시켜 확장시킨다. 확장된 메시지-순차적 다이어그램은 테스트케이스가 발생 가능한 원인-결과 다이어그램과 접목시켰다. 접목한 원인-결과 다이어그램을 통해서 테스트케이스 추출하였다.