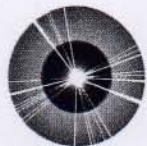


제13권 제1호
Vol. 13 No. 1



한국정보과학회
Korean Institute of Information Scientists and Engineers



한국정보처리학회
Korean Information Processing Society



2011

한국 소프트웨어공학 학술대회 논문집(안)

(Preliminary)

Proceedings of 2011 Korea Conference on
Software Engineering

- 일시 : 2011년 2월 9일(수)~11일(금)
- 장소 : 한화 휘닉스 파크(강원도 평창)

주최 : 한국정보과학회, 한국정보처리학회

주관 : 한국정보과학회 소프트웨어공학 소사이어티
한국정보처리학회 소프트웨어공학 연구회
한국전자통신연구원

후원 : 삼성 SDS, 비트컴퓨터, (주)모아소프트, (주)이웨이파트너즈,
단국대학교SERC(소프트웨어공학연구센터), 고려대학교 고신뢰
융합소프트웨어 연구센터, 서강대학교 SW 요구 및 검증공학기술
연구센터, 숭실대학교 모바일서비스 SW공학센터, 포항공대
융합소프트웨어개발 연구센터

모바일 애플리케이션의 보안취약성 검사 도구 설계-----	497
최윤희, 최은만 (동국대학교)	
스마트폰앱의 호환성 테스트 기법-----	505
이정욱, 김태연, 채흥석 (부산대학교)	
xUnit 을 이용한 모바일 애플리케이션의 테스트 패턴-----	513
마영철, 최은만 (동국대학교)	

G3: 소프트웨어 프로세스 및 관리 II

소프트웨어 글로벌화 품질 최적화 방안 - WBS 과제 적용 사례 중심-----	525
이세영, 이상은, 이혁재 (정보통신산업진흥원)	
가상 환경상에서 다관절 로봇의 테스트프로세스 연구-----	532
손현승, 김우열, 김재승, 김영철 (홍익대학교)	
전문적인 지식의 효과적인 커뮤니케이션을 위한 소프트웨어 기술문서 작성 가이드---	536
전수남 (상명여자대학교)	

포스터

리눅스 기반 이기종 무선 망간 자동 핸드오버 모델 설계 및 구현-----	549
김동용, 이우승, 제은경, 홍지만 (송실대학교)	
해체작업 단위생산성 산출 시스템-----	553
박승국, 지연희, 문제권 (한국원자력연구원)	
ERP 기반의 도로건설공사 현장 업무 프로세스 재설계 방안 연구-----	557
정성윤, 서명배, 나혜숙, 최원식 (한국건설기술연구원)	
고장 감내형 임베디드 프로세서 검증-----	561
고완진, 고규천, 이슬기, 황용완, 나종화 (한국항공대학교)	

가상환경상에서 다관절로봇의 테스트 프로세스 연구

손현승, 김우열, 김재승, 김영철

홍익대학교 컴퓨터정보통신공학과
 소프트웨어공학전공
 {son, john, kjs, bob}@selab.hongik.ac.kr

요약: 기존의 테스트는 설계단계에서 미리 테스트 케이스를 생성하지만 실제 수행은 구현이 된 후에 가능하다. 그만큼 테스트 설계와 실행의 시간차가 많이 발생하여 테스트 설계 및 소프트웨어 설계의 오류를 확인하기 어렵다. 본 논문에서는 설계 단계에서 테스트 케이스를 생성하여 가상의 환경에서 바로 실행 가능한 프로세스를 제안한다. 제안한 프로세스는 다관절로봇 개발 방법에 테스트과정을 포함한다. 테스트과정은 모델로부터 테스트케이스를 만들고 가상의 시뮬레이터에서 수행 가능하다. 이를 통해서 테스트 설계와 실행의 시간차를 줄여 테스트 케이스의 오류를 쉽게 발견하고 설계상에서 발생하는 문제를 빠르게 대처 가능하다. 소프트웨어의 개발 초기 단계에서 오류를 식별하고 검증함으로써 개발에 필요한 시간과 비용을 줄일 수 있다.

핵심어: 다관절로봇, Modeling & Simulation, 테스트, 가상환경, 모델기반테스트

1. 서론

1968년 McGhee와 Frank[1]가 최초로 컴퓨터 제어 자동로봇을 고안한 이후에, 다관절로봇[2]은 일족에서 다족의 여러 형태로 제작되었다. 이전에는 다리의 기계적인 메커니즘으로 단순 동작이 수행되었고, 최근에는 다양한 동작을 위해 다리의 관절에 액추에이터를 달아 모두 제어 한다. 이렇게 다양한 동작 수행으로 물건 옮기기, 계단 오르기 등과 같이 복잡한 작업이 가능해졌고 교육, 엔터테인먼트, 군사용 등 그 활용도가 매우 높아 졌다. 많은 관절 및 다수의 다리 제어를 위해 복잡한 처리를 수행하는 소프트웨어가 필요하다. 예를 들어 소형 다관절로봇인 6족로봇은 3개의 관절에 총 18개의 모터를 동시에 제어해야 한다. 이러한 환경에서 동작은 그만큼 정교

한 제어가 필요한 것이다[3].

이러한 이유로 다관절로봇 개발에서는 복잡한 동작생성을 위해 Modeling & Simulation을 사용한다 [4,5,6]. M&S 기반의 개발은 다관절로봇의 형태를 조합하고 행위를 생성하는 Pre-Modeling 단계와 가상의 환경에서 조립된 로봇에 모션데이터를 삽입하여 수행하는 Pre-Simulation 단계로 구분된다. 이 두 단계를 통해서 가상의 환경에서 로봇의 동작상태를 점검한다. 이후에 로봇의 소프트웨어를 설계, 구현, 테스트과정을 거친다. 그러나 기존의 다관절로봇 개발에서의 테스트는 최종의 구현 후에 테스트 수행이 가능하다.

소프트웨어 개발에서 오류를 어느 단계에서 발견하느냐에 따라서 오류 해결을 위한 비용이 크게 차이가 난다. 요구 단계에서 오류를 발견하여 해결하는데 드는 비용이 1이라고 한다면 제품생산 단계에서 오류를 발견하고 이를 해결하는데 드는 비용은 40~1000 배 정도 많이 든다[7]. 그렇기 때문에 소프트웨어의 테스트를 구현 후 수행하는 것이 아니라 이전 단계에 수행한다면 소프트웨어의 개발 초기 단계에서 오류를 식별하고 검증함으로써 개발에 필요한 시간과 비용을 줄일 수 있다. 또한 설계와 구현 단계의 테스트 수행으로 좀더 높은 품질의 소프트웨어를 보장할 수 있다.

본 논문에서는 가상환경에서의 테스트 프로세스를 제안한다. 제안한 테스트 프로세스는 요구사항 분석, Pre-모델링, Pre-시뮬레이션, 설계, 구현, 모델기반테스트, 코드기반테스트 총 7단계의 절차로 구성된다. 기존에는 설계 단계에서 테스트케이스를 생성하지만 실제 수행은 구현이 된 상태에서 수행된다. 그만큼 테스트 설계와 실행의 차이가 생긴다. 그러나 본 방법은 가상의 환경에서 직접 테스트 케이스를 실행 가능하여 구현단계까지 기다리지 않더라도 테스트를 수행하여 설계의 오류를 미리 검증할 수 있다.

본 논문의 구성은 다음과 같다. 2 장에서는 관련된 구로서 M&S 기반 다관절로봇 개발 방법과 기존의

[†] 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업(2010-0012117)과 교육과학기술부와 한국연구재단의 지역혁신인력양성사업으로 수행된 연구결과임.

테스트 프로세스 및 모델기반테스트에 대해서 기술한다. 3 장에서는 제안한 테스트 프로세스에 대해서 정의한다. 4 장에서는 제안한 테스트 프로세스의 설계 및 모델기반테스트를 적용사례를 통해 보여준다. 마지막으로 5 장에서는 결론 및 향후 연구에 대해서 언급한다.

2. 관련연구

2.1 M&S 기반 다관절로봇 개발 방법

로봇 개발환경은 크게 미들웨어를 이용한 방법, 미들웨어를 사용하지 않은 펌웨어 기반으로 분류된다. 보통 미들웨어를 이용한 로봇은 커다란 시스템에서 사용되고 펌웨어를 이용한 개발은 작은 로봇에서 사용된다. 미들웨어는 시스템과 응용프로그램간의 중간에 위치하여 상호운영성 및 개발 편의성을 높여주지만 실행속도가 떨어지고 하드웨어의 높은 성능을 요구하기 때문에 작은 시스템에서 사용하기 어렵다. 본 논문에서 제안하는 방법은 미들웨어를 쓰지 않으면서 개발할 수 있는 방법으로 작은 시스템부터 큰 시스템까지 개발 가능하다. 또한 기존의 개발 방법들은 로봇에 대한 모델링 및 시뮬레이션에 대해 제공하지 않아 개발이 어렵다. M&S 기반 개발환경은 다관절로봇에 최적화된 시뮬레이션을 제공하여 로봇을 빠르게 개발 가능하다[6].

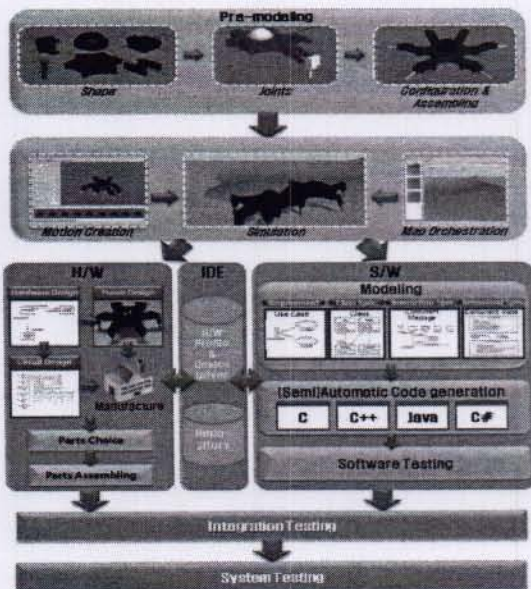


그림 1. M&S 로봇 개발 환경

로봇 개발 환경은 크게 Pre-Modeling 단계, Pre-Simulation 단계, 소프트웨어 개발 단계, 하드웨어 개발 단계, 통합 테스트 단계로 구성된다. 그림 1 은 개발 환경을 도식화 한 것이다.

2.2 기존의 테스트프로세스

V-모델은 개발 프로세스와 테스트 프로세스를 잘 매칭 하고 있다. V-모델에서는 폭포수 모델에 시스템의 각 단계를 정확히 수행하고 있는 확인하는 validation 과 테스트 작업을 강조한 모델이다[8]. V-모델은 기존의 폭포수 모델의 단점을 보완하고 있다. V 모델은 그림 2 에서처럼 개발과 테스트가 좌우 대칭을 이루고 있다. 또한 테스트 프로세스와 개발 프로세스를 잘 매칭하고 있고 실제 테스트에서 각 단계에서의 활동은 문서적으로 확인하거나 리뷰하는 방법을 제시하고 있다.

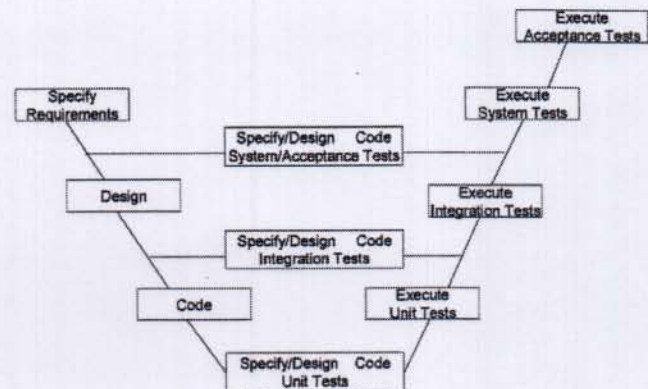


그림 2 기존의 V-Model

V-모델은 단위, 통합, 시스템 이렇게 3 가지 테스트 레벨을 가지고 있다. 이것은 V-모델 뿐만 아니라 테스트 프로세스에는 기본적으로 포함되어 있는 개념이다.

단위 테스트는 소프트웨어 설계의 최소단위인 모듈을 검증하는 것이다. 이러한 모듈검증에 초점을 맞추어 상세 설계의 지침으로 중요한 제어 경로를 테스트한다. 단위 테스트의 절차를 살펴보면, 모듈은 독립된 프로그램이 아니기 때문에 테스트 드라이버 또는 스텝을 만든다. 단위 테스트 환경 대부분의 드라이버는 모듈로 전달하며 관련된 결과를 출력하는 단순한 역할을 담당하는 주 프로그램의 기능을 갖는다.

통합 테스트는 모듈을 몇 개 결합하여 하나의 시스템으로 완성시키는 과정에서의 테스트이다. 이 과정은 각 모듈간의 인터페이스를 검사하게 된다. 통합 테스트에서 주로 사용하는 방법에는 하향식 테스트와 상향식 테스트가 있다. 하향식 테스트는 시스템의 최상위에 있는 주 프로그램 모듈을 먼저 테스트하고 점차 하위 레벨로 테스트를 이동하는 방법이다. 상향식 테스트는 시스템 하위 레벨의 모듈로부터 테스트를 진행하는 경우로 점차 상위레벨로 이동한다.

시스템 테스트는 서로 다른 목적을 가진 일련의 테스트들로 이루어지는 시스템에 관한 테스트이다. 시스템의 모든 요소들이 적합하게 통합이 되고 올라

른 기능을 수행하는지를 검사한다.

3. 가상환경에서의 테스트 프로세스

제안한 테스트 프로세스는 요구사항 분석, Pre-모델링, Pre-시뮬레이션, 설계, 구현, 모델기반테스트, 코드기반테스트 총 7 단계의 절차로 구성된다.

요구사항 단계(Step. 1)에서는 로봇에 대한 요구사항을 추출하고 분석한다.

Pre-모델링 단계(Step. 2)에서는 타겟 오브젝트를 선택하여 조립하고 가상객체의 모션 추출을 수행한다. 타겟 오브젝트의 선택은 로봇의 모델을 구성하기 위해서 부품들을 조립하는 것이다. 타겟 오브젝트 조립에서는 선택된 부품들을 이용하여 원하는 로봇으로 조립한다.

Pre-시뮬레이션 단계(Step. 3)에서는 Pre-모델링 단계에서의 로봇 모델을 가상 환경상에서 시뮬레이션을 수행한다. 이 단계에서는 로봇이 수행될 곳의 환경을 분석하여 가상이지만 실제처럼 꾸며 로봇이 임무를 수행하면서 발생하는 문제를 분석할 수 있는 기반을 마련한다.

설계 단계(Step. 4)에서는 로봇의 모델을 이용하여 모델을 구축한다. 이때 클래스, 시퀀스, 상태 다이어그램을 이용하여 로봇을 설계한다. 로봇의 부품은 각각의 클래스로 변환 할 수 있다. 이때 로봇의 부품이 가져야 될 역할 및 속성, 행위를 결정한다.

모델기반테스트(Step. 5)는 설계 단계에서 생성된 모델을 기반으로 테스트 케이스를 생성하고 M&S 환경상에서 테스트 수행한다. 모델기반의 테스트 케이스 생성은 설계 단계에서 생성된 모델을 구현 전에 가상의 환경에서 테스트하기 위해서 테스트 케이스를 생성한다.

구현 단계(Step. 6)은 설계 모델을 기반으로 로봇이 수행할 수 있는 코드를 생성하고 구현한다. 다이어그램 메타 모델과 코드 생성 템플릿을 이용하여 대상 코드를 생성한다.

코드기반테스트(Step. 7)은 구현단계에서 생성된 코드를 기반으로 화이트 박스 테스트를 수행한다. 테스트 케이스 생성은 기존의 테스트 방법과 동일하다. 모델기반테스트와 코드기반테스트를 모두 수행함으로써 지능형 로봇 소프트웨어를 구현할 때 보다 높은 품질을 향상시킬 수 있다.

본 논문에서는 제안한 테스트 프로세스 중에서 설계단계(Step. 4), 모델기반테스트(Step. 5)에 대해서 중점적으로 다룬다.

4. 설계 및 모델기반테스트

4.1 설계

예를 들어 다관절로봇이 전, 후, 좌, 우의 4 가지 방

향을 이용해서 목표물 지점까지 이동한다면 다관절 로봇은 그림 3 과 같이 상태 다이어그램을 작성 한다. 'Idle'은 로봇의 초기 상태로 처음 시작하거나 모든 작업이 끝난 로봇의 상태가 이 상태에 해당된다. 'Initialized'는 로봇이 초기화되는 상태를 나타낸다. 이 상태에서 로봇의 목표를 지정하고 좌표를 설정하게 된다. 'goRobot'은 정해진 위치로 로봇이 이동하는 상태를 나타낸다. 이 상태에서는 로봇이 앞 또는 뒤로 전진하면서 현재 위치를 체크하게 된다. 'goLeft'는 로봇이 좌측으로 회전하는 상태이다. 'goRight'는 로봇이 우측으로 회전하는 상태이다. 'Stopped'는 사용자가 로봇을 정지시켰을 때 발생하는 상태이다. 'Stopped by Intrusion'은 로봇이 목적지와 도착하였을 때 발생하는 상태이다. 로봇은 시작 시 'Idle' 상태를 가지게 되며, 사용자가 로봇의 정지를 명령했을 때 'Idle' 상태로 전환 또는 완전히 정지를 하게 된다.

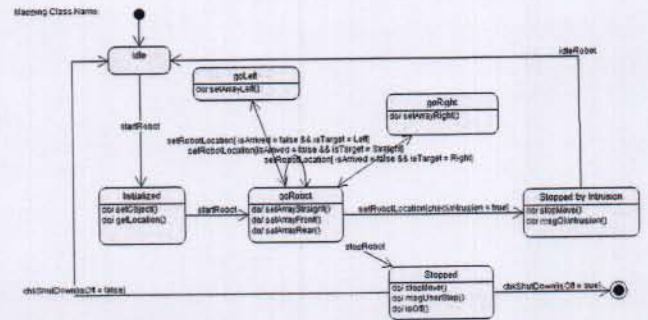


그림 3 다관절로봇의 상태 다이어그램

4.2 모델 기반 테스트

테스트를 수행하기 위해서 테스트 케이스를 자동 생성해 준다. 테스트 케이스를 생성하기 위해서는 설계도구를 통해 상태 다이어그램을 작성해야 하고 상태 테이블을 생성한다. 상태 테이블이 생성된 후 현재 상태 테이블을 토대로 하여 상태 전이 트리를 생성한다. 상태 전이 트리는 현재 상태에서 호출할 수 있는 모든 상태를 트리 형식을 통해 그려내 눈으로 확인할 수 있게 해준다. 트리의 가지는 깊이가 깊어 질수록 더욱 많은 가지 수를 가지게 되며, 이러한 깊이는 Switch 를 선택함으로써 정해줄 수 있다.

상태 전이 트리가 생성되면 최종적으로 테스트 케이스가 생성된다. 테스트 케이스를 통해 현재 상태(Start State)에서 어떠한 이벤트와 어떠한 액션이 발생할 수 있는지를 알아낼 수 있으며, 이를 통해 다음 상태(Next State)가 어떠한 상태가 되는지를 테이블 형식으로 확인할 수 있다.

최종적으로 생성된 테스트케이스를 수행하기 위해서는 로봇 시뮬레이터와 설계도구를 연동하고 설계 도구 내에서 Event List 에 테스트 케이스를 등록한다. 실행 버튼을 클릭하면 상태다이어그램의 상태들을 수행하면서 시뮬레이터의 로봇을 수행하게 된다.

5. 결 론

소프트웨어 개발에서 오류를 어느 단계에서 발견하느냐에 따라서 오류 해결을 위한 비용이 크게 차이가 난다. 그렇기 때문에 소프트웨어의 테스트를 구현 후 수행하는 것이 아니라 이전단계에 수행한다면 소프트웨어의 개발 초기 단계에서 오류를 식별하고 검증함으로써 개발에 필요한 기간과 비용을 줄일 수 있다. 그러나 기존의 테스트 방법은 구현이 이루어진 다음에 테스트를 수행할 수 있어 테스트케이스 설계와 실행의 기간의 차이가 생긴다.

본 논문에서는 이러한 문제를 해결하기 위해 가상 환경에서의 테스트 프로세스를 제안하였다. 제안한 테스트 프로세스에는 상태 다이어그램을 기반으로 하여 테스트케이스를 생성하는 방법과 가상의 환경에서 생성한 테스트케이스를 수행하는 구조를 제안하였다. 이 방법을 다관절로봇에 테스트 케이스 발생 및 수행을 적용해 보았다. 그 결과 설계 단계에서 생성한 테스트 케이스를 가상의 환경에서 직접 수행할 수 있었다.

현재의 본 논문의 방법에서는 가상의 시뮬레이션 환경에서 수행된 테스트케이스를 테스터가 직접 눈으로 확인하고 체크해야 한다. 이를 극복하기 위해서 향후 연구로 테스트 기대결과에 대한 자동확인을 연구 중이다.

참고문헌

- [1] R.B. McGhee, A.A. Frank, "On the Stability Properties of Quadruped Creeping Gaits", *Mathematical Biosciences*, Vol. 2, No 1/2, 1968.
- [2] Marc H. Raibert, "Legged Robots", *ACM*, Volume 29, Issue 6, pp. 499-514, 1986.
- [3] 손현승, 김우열, 김영철, "다관절 로봇의 효율적인 동작제어를 위한 기술 구현", *한국정보처리학회*, Vol. 15, No. 2, pp. 593-596, 2008.11.14.
- [4] Woo Yeol Kim, Hyun Seung Son, R. Young Chul Kim, C. R. Carlson, "MDD based CASE Tool for Modeling Heterogeneous Multi-Jointed Robots", *CSIE 2009, IEEE Computer Society, Los Angeles/Anaheim, USA*, Vol. 7, pp. 775-779, 2009.04.01.
- [5] 김재수, 손현승, 김우열, 김영철, "다관절 로봇 제어를 위한 교육용 소프트웨어 연구", *한국정보교육학회*, Vol. 12, No. 4, pp. 469-476, 2008.12.
- [6] 김재수, 손현승, 김우열, 김영철, "자율탐색 로봇 설계를 위한 M&S(Modeling & Simulation) 환경 연구", *한국군사과학학회*, Vol. 11, No. 6, pp. 127-134, 2008.12.

- [7] Boehm, B. W., *Software Engineering Economics*, Prentice-Hall, Englewood Cliffs, NJ., 1981.
- [8] Ilene Burnstein, *Practical Software Testing*, Springer-Verlag, 2003.