

# 자율탐색 로봇 설계를 위한 M&S(Modeling & Simulation) 환경 연구

A Study on M&S Environment for Designing the Autonomous  
Reconnaissance Ground Robot

**김재수\***  
Kim, Jae-Soo

**손현승\*\***  
Son, Hyun-Seung

**김우열\*\***  
Kim, Woo-Yeol

**김영철\*\***  
Kim, R. Young-Chul

## ABSTRACT

An autonomous reconnaissance ground robot performs its duty in various different environments such as mountain-scape, desert and under-water through changing its shape and form according to the environment it is working in. Making a prototype robot for each environment requires extra cost and time. It is also difficult to modify the problem after production. In this paper, we propose the adoption of M&S(Modeling & Simulation) environment for the production and design of the autonomous reconnaissance ground robot. The proposed method on the M&S environment contributed to the more effective and less time consuming production of the robot through the Pre-Modeling and Pre-Simulation process. For example, we showed the design and implementation of the autonomous reconnaissance ground robot under the proposed environment and tools.

주요기술용어(주제어) : Modeling & Simulation(모델링과 시뮬레이션), Robot Development Tool(로봇 개발 도구), Autonomous Reconnaissance Ground Robot(자율 탐사 로봇), Robot Design(로봇 설계), Robot Development Environment(로봇 개발 환경)

## 1. 머리말

최근 미국에서는 군사용으로 6족로봇 Rhex, Sprawlettes, 휴대용 무인지상차량(UGV), 의료용 로

봇, 4족 견마형 로봇, 곤충형 로봇, 시큐리티/조사 로봇 등을 DARPA(Defense Advanced Research Projects Agency)에서 지원해주고 대학 중심으로 개발하고 있다<sup>[1]</sup>. 이러한 로봇들은 다양한 환경에서 동작하기 위해 특이한 하드웨어 구조를 가진다. 주로 사용되는 하드웨어는 2~6바퀴를 사용하는 로봇, 상단, 하단 체인을 사용하는 로봇, 2~6개의 다리를 사용하는 로봇 등 이동수단에 의하여 종류가 구분된다.

자율탐색 로봇은 산악지형, 사막, 수중과 같은 다양한 환경 속에서 임무를 수행하게 된다. 자율탐색 로

† 2008년 10월 13일 접수~2008년 11월 28일 게재승인

\* 한국과학기술정보연구원(Korea Institute of Science and Technology Information)

\*\* 홍익대학교(Hongik University)

교신저자 이메일 : bob@hongik.ac.kr

봇의 종류는 우주 탐사, 위험 탐사, 파이프 탐사, 화산 탐사, 수중 탐사와 같이 탐사 지역에 따라 그 로봇의 형태와 모양이 달라진다. 화성탐사 로봇 소저너 같은 경우는 6개의 바퀴로 구성되어 있고 로봇팔과, 태양전지가 탑재되어 있는 특징이 있다. 위험 탐사로봇은 주로 체인형태의 바퀴가 주로 사용된다. 화산 탐사로봇은 6족형 다리가 사용되는 관절형 로봇이 사용된다. 수중 탐사 로봇은 물속에서 이동을 위해 프로펠러가 장착되어 있다.

자율탐색을 위해서는 각 환경에 알맞은 로봇을 제작해야 된다. 이때 환경에 맞는 로봇을 만들기 위해서 프로타입형 로봇을 제작하게 되면 별도의 비용과 시간이 추가로 소모되며 문제점을 발견하였을 때 수정이 어렵다. 그래서 자율탐색로봇을 미리 만들어보고 이를 시물레이션 할 수 있는 모델링 & 시물레이션이 필요하다. 본 논문에서는 이러한 문제점을 해결하고자 로봇 개발방법론에 모델링과 시물레이션 과정을 추가하여 실제 로봇 없이도 쉬운 동작을 만들 수 있도록 개발 환경과 도구를 지원한다. 그 뿐만 아니라 이종의 기기와 하드웨어의 개발 없이 로봇을 개발할 수 있는 방법론<sup>[2~4]</sup>을 포함한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구로서 로봇 개발 환경에 대하여 설명한다. 3장에서는 제안한 모델링 시물레이션 개발 환경과 도구에 대하여 언급한다. 4장에서는 적용사례로서 제안한 개발 환경으로 6족로봇을 자동탐사로봇으로 구현해 본다. 5장은 제안한 방법의 개발 시간을 측정하여 분석한다. 마지막 장에서는 결론 및 향후 연구를 언급한다.

## 2. 관련 연구

기존 자율 로봇의 M&S 환경을 살펴보면 표 1과 같이 CAMUS, OSGI, ER, MSRS 등이 있다.

CAMUS(Context-Aware Middleware for URC Systems)는 URC 환경 내에서 획득된 상황 정보를 기반으로 환경 내에 있는 사용자에게 적절한 서비스를 제공할 수 있도록 상황 기반 응용의 개발과 실행을 지원하는 상황 인식 미들웨어이다<sup>[5]</sup>. CAUMUS는 ETRI에서 개발한 자바 기반의 URC 로봇 개발 플랫폼

으로 자바 기반의 라이브러리 및 연동 모듈 제공하는 장점을 가지지만 시물레이션 환경 지원 안 되고 일반인들이 다운로드 받아 사용하거나 개발하기 어렵다.

[표 1] 로봇 개발 환경 비교 표<sup>[9]</sup>

구분	CAMUS	OSGI	ER	MSRS
소개	ETRI의 URC 로봇 개발 플랫폼	U-Home 솔루션에 적용	SDK 기반의 상용제품	MS 로봇개발용 플랫폼
장점 및 특징	자바 기반의 라이브러리 및 연동 모듈 제공	자바 기반의 API 및 실행 플랫폼으로 구성	컴퓨터 비전, 자율 내비게이션 SW기술 제공	로봇 업계에 있어서 공통화된 표준 플랫폼을 제공함
단점	일반인들이 사용하기 어려움	일반인들이 사용하기 어려움	제품 사용하기 위해 비용 지불	무선 통신 요구
시물레이션 환경	지원 안됨	지원 안됨	지원 안됨	지원됨

OSGI(Open Services Gateway Initiative)는 1999년에 썬 마이크로시스템즈, IBM, 에릭슨 등에 의해 구성된 개방형 표준 단체이다. 그 후로 OSGI는 원격 관리 될 수 있는 자바 기반의 서비스 플랫폼을 제정해왔다<sup>[6]</sup>. 이 표준 사양의 핵심은 어플리케이션의 생명주기모델과 서비스 레지스트리를 정의하는 프레임워크이다. 자바 기반의 서비스 플랫폼으로 U-Home 솔루션에 적용되어, 가정의 가전기기의 원격 제어 및 연동에 활용되고 있다. 자바 기반의 API 및 실행 플랫폼으로 구성되어 있으며 시물레이션 환경이 없는 것이 단점이다. 또한 일반인들이 다운로드 받아 사용하거나 개발하기가 어려움을 가지고 있다.

ER(Evolution Robotics)은 로봇 제품 제작을 위한 종합적 개발 플랫폼으로 개발자들이 강력한 애플리케이션을 신속하고 용이하게 구축할 수 있게 하는 중요한 기반구조, 핵심 능력 및 도구를 제공한다<sup>[7]</sup>. SDK 기반의 제품으로서 상용제품으로 컴퓨터 비전, 자율 내비게이션, 인간-로봇 인터랙션 및 시스템 아키텍처를 위한 소프트웨어 기술을 제공한다. 단점으로는 시

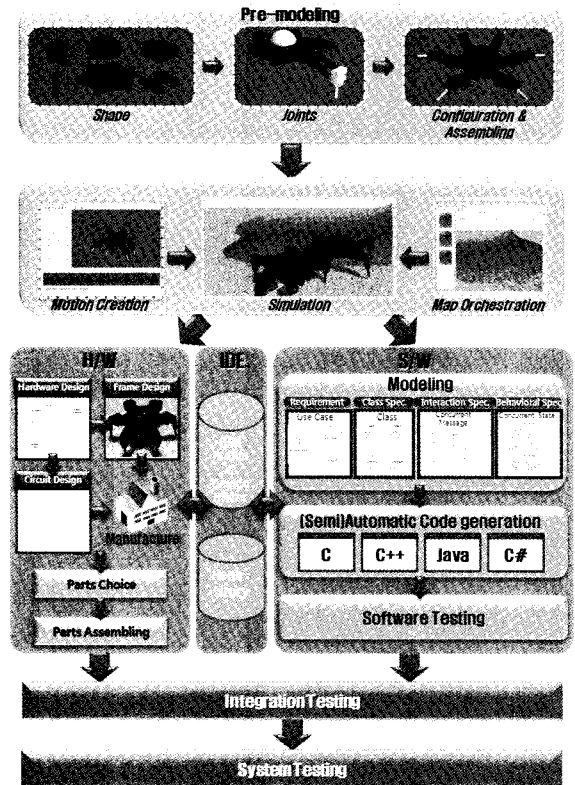
물레이션 환경이 지원되지 않고 상용 제품이기 때문에 추가 개발비용이 든다.

MSRS(Microsoft Robotics Studio)는 로봇분야에 관심이 있는 개발자, 학생 또는 일반인들이 프로그래밍에 대한 기본적인 지식만 가지고 있으면, 다양한 로봇 하드웨어 상에서 필요로 하는 다양한 로봇 애플리케이션을 쉽게 개발할 수 있도록 지원하는 개발 툴 및 환경이다<sup>[8]</sup>. 다양한 컴퓨터 기기들 간의 Connectivity가 제공되고 스마트폰에서 부터 PDA, Windows CE, UMPC, Windows XP, Windows Vista, Windows Server 2003 등을 지원한다. 또한 다양한 개발자 커뮤니티와 두터운 개발자 층을 확보하고 깊이 있는 개발기술 제공한다. 시뮬레이션 환경을 지원하여 H/W 로봇이 없이도 로봇 프로그램이 가능하다. 단점으로는 MS 윈도우 제품에서만 동작되고 Connectivity가 강조되어 로봇이 무선통신장비가 탑재되어 있어야 한다. 그리고 높은 하드웨어 사양을 요구한다.

### 3. 제안한 M&S 환경

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 상호운용성 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 개발가능하다. 또한 기존의 개발 방법들은 로봇에 대한 모델링 및 시뮬레이션에 대해 제공하지 않아 개발이 어렵다. 본 논문의 개발환경은 다관절로봇에 최적화된 시뮬레이션을 제공하여 로봇을 빠르게 개발가능하다.

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



[그림 1] M&S 로봇 개발 환경

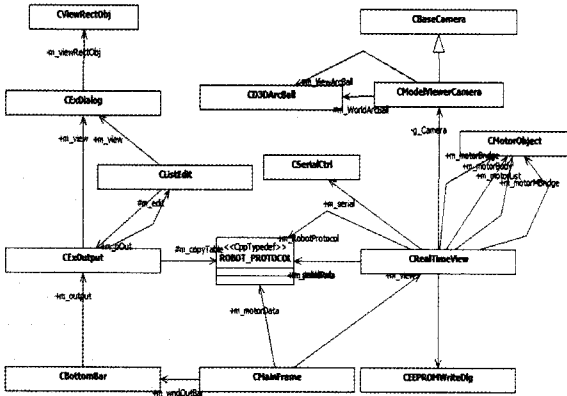
Pre-Modeling과 Pre-Simulation을 수행하기 위해서는 이를 지원해주는 도구가 필요하다. 그래서 본 절에서는 모델링과 시뮬레이션을 수행할 수 있게 지원해주는 모션 생성 도구와 시뮬레이션 도구에 대해서 설명 한다.

#### 가. Pre-Modeling

Pre-Modeling 단계는 로봇의 모양을 모델링하고 그 부품을 조인트를 이용하여 구성시키고 모델을 만들어 내는 단계이다. 이 때 로봇에 센서와 같은 하드웨어 장비도 구성하여 실제 개발할 로봇과 똑같은 형태로 만들게 된다. 이 단계를 지원해주는 도구는 모션 생성 도구이다.

모션 생성 도구의 역할은 여러 개로 구성된 모터들의 각도 값을 쉽게 생성하여 다관절 로봇의 동작을 빠르게 개발할 수 있도록 지원해 주는 것이다. 모션 생성 도구는 실제 모양과 같은 3D형태의 로봇을 마우

스로 움직여 그 각도를 생성해 주는 것이다. 그래서 실제 로봇이 없더라도 모션 데이터를 생성 해낼 수 있다. 모션 생성도구의 전체 구조는 그림 2와 같다.



[그림 2] 모션 생성 도구의 클래스 다이어그램

개발환경은 C++이용하여 DirectX 9.0 SDK와 MFC(Microsoft Foundation Class)가 사용되었다. CRealTimeView는 로봇을 렌더링 하고 마우스입력을 데이터 모델에 주는 역할을 한다. CModelViewCamera는 마우스 오른쪽 버튼을 입력하였을 때 카메라를 회전시켜 3D모델을 회전시키는 역할을 한다. 그리고 CExOutput은 도구 오른쪽에 있는 다이얼로그 바로 모터 값을 수치를 입력할 수 있도록 하여 미세 조정이 가능하게 한다. 그리고 CSerialCtrl은 로봇과 통신을 위해 시리얼 통신을 제어하는 부분이다.

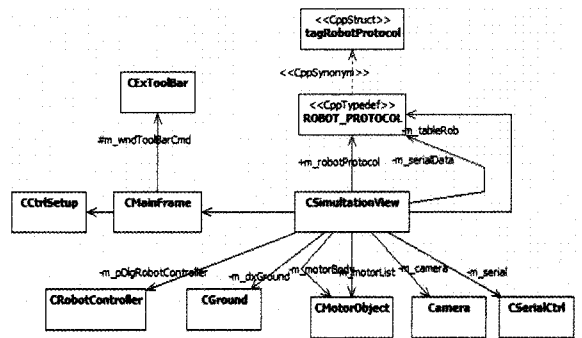
```
typedef struct tagRobotProtocol
{
    BOOL bWrite;
    BYTE mode;
    BYTE motor[18];
    BYTE special[6];
    BYTE speed;
    BYTE delay;
    CString comment;
}ROBOT_PROTOCOL;
```

[그림 3] ROBOT\_PROTOCOL의 구조체

로봇을 실시간으로 제어하기 위해서 ROBOT\_PROTOCOL이 사용되는데 그 구조는 그림 3과 같다. bWrite는 초기화된 이후로 모션 값이 새롭게 쓰였는지 확인하는 변수 이다. mode는 로봇의 동작을 나타내는 것으로 실시간 전송 모드, 자율 주행모드, 데이터 저장 모드로 구분된다. motor는 실제 모터의 각도 값이고 special은 확장용으로 미리 할당해둔 것이다. speed는 속도, delay는 지연시간, comment는 설명 값을 저장한다.

#### 나. Pre-Simulation

Pre-Simulation단계는 모델링 단계에서 생성한 모델을 이용하여 시뮬레이션을 수행 한다. 이때 맵 에디터를 이용하여 여러 가지 환경을 만들어 시뮬레이션에 활용할 수 있다. 모델링과 시뮬레이션 단계를 미리 진행하여 로봇의 움직임이나 로봇의 구조를 실제 로봇을 만들기 전에 테스트 하여 문제점이나 요구사항을 미리 발견할 수 있다. 그리고 하드웨어 없이 로봇 소프트웨어를 개발할 수 있기 때문에 소프트웨어 중심으로 하드웨어 개발을 진행시킬 수 있다. 이 단계에서는 시뮬레이션 도구가 제공된다.



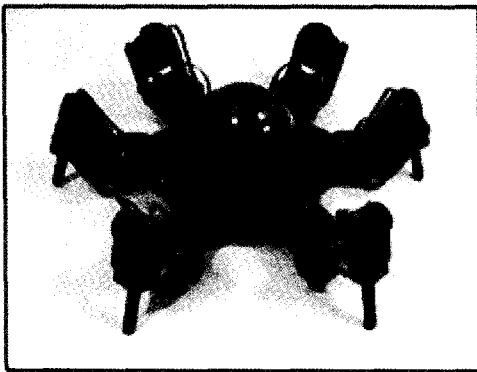
[그림 4] 시뮬레이션 도구의 클래스 다이어그램

시뮬레이션 도구는 모션 생성도구로 부터 생성된 로봇 행위를 입력 받아 가상의 환경 상에서 수행하는 것을 말한다. 시뮬레이션은 동역학 시뮬레이션을 위해 ODE(Open Dynamic Engine)<sup>[10]</sup>를 사용한다. ODE는 강체 역학 시뮬레이션 엔진으로 로봇을 시뮬레이션하기에 적합한 형태를 가지고 있다. 그림 4는 시뮬레이션 도구의 클래스 다이어그램이다.

ROBOT\_PROTOCOL은 로봇 행위 데이터 파일을 해석한다. 그리고 CSimulationView는 로봇 모델과 가상 환경을 렌더링하고 로봇의 모션데이터를 처리한다. CRobotController는 로봇 행위 데이터가 없을 경우 미리 지정된 데이터를 통해 로봇을 제어할 수 있도록 한다. 그리고 CGround는 가상의 환경 파일을 처리하고 Camera는 화면 카메라를 제어한다. CExToolBar는 모션의 시작, 일시정지, 멈춤을 동작을 제어할 수 있다. CSerailCtrl은 로봇과 실시간통신을 할 수 있도록 데이터를 전송해 준다.

#### 4. 적용 사례

자율탐색 로봇에 개발 환경을 적용시키기 위해서 그림 5와 같은 6족형 로봇을 사용하였다. 본 예제의 6족 로봇은 한 다리 당 3개의 관절을 사용하여 총 18개의 모터를 사용하는 구조로 되어 있다. 그리고 다리가 원통형으로 균일하게 분배되어 있어 앞뒤 좌우에 상관없이 몸을 회전하지 않고 로봇을 이동시킬 수 있는 특징을 가진다.



[그림 5] 6족형 자율탐색 로봇

##### 가. 자율탐색로봇 구성

자율탐색로봇의 구성은 표 2와 같다. 6족형 자율탐색로봇은 모터제어부에 Atmega 128를 센서 제어부에 Atmega 8583을 사용한 2개의 마이크로컨트롤러를 가지고 있다. 그리고 4축 방향으로 초음파 센서가 장착되어 사방의 물체를 감지할 수 있도록 되어 있다. 단

반향 자이로센서 2개가 장착되어 x축, y축 방향의 기울기를 탐지할 수 있다. 또한 다리당 3개의 관절이 사용되어 총 18개의 서보모터를 사용한다. 실시간제어를 위해 블루투스가 사용되었다.

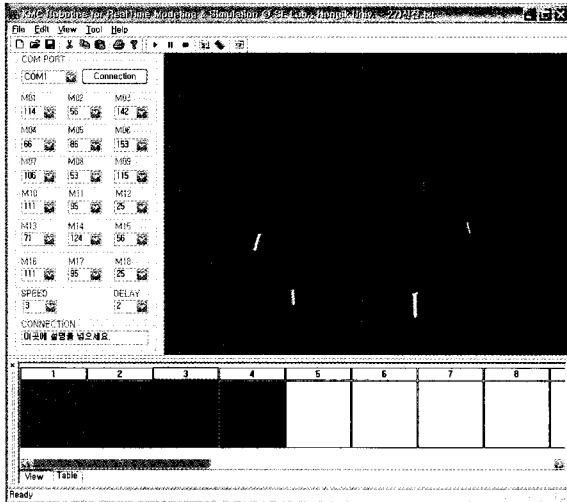
[표 2] 자율 탐색 로봇 하드웨어 구성표

구분	자율 탐색 로봇	
Microcontroller	Main	Atmega 128 16MHz
	Sub	Atmega 8583 16MHz
RAM	Main	4KByte
	Sub	512Byte
EEPROM	Main	128KByte
	Sub	8KByte
OS	none	
Sensor	4	
Gyro Sensor	2	
Communication	Bluetooth	
Motors	18	
Languages	C	

##### 나. 로봇 모션 생성

모션 생성도구는 가상으로 모델링된 3D 모델을 마우스 조작으로 각 모터의 각도 값을 기록하였다가 로봇의 순간 동작을 만들어내는 도구이다. 그림 6은 모션 생성 도구를 이용하여 행위 데이터를 생성하는 도구의 모습이다. 앞으로 가는 동작을 만들기 위해서 가만히 서있는 동작에서 로봇 모터 M4M5M6, M7M8M9를 후방 대각선 45도로 모아주고 M13M14M15, M16M17M18을 전방 대각선 45도 모아준다. 이렇게 하면 M13M14M15와 M1M2M3사이에 60도 이상의 공간이 생겨 M16M17M18이 전방으로 다리를 뻗어 자유롭게 움직일 수 있다. 그 다음 M7M8M9의 다리를 아래로 내리면서 M16M17M18을 앞쪽으로 들어 M1M2M3다리가 가까이 놓는다. 그러면 앞의 다리가 앞으로 끌어당기면서 로봇이 앞으로 이동한다. 그다음 뒤에 있는 M7M8M9다리를 M10M11M12가 가까이 옮

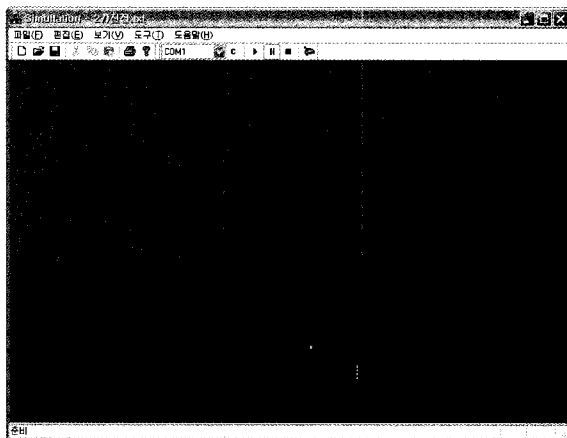
김으로 처음 동작의 반대쪽의 각의 형성을 이루면서 M1M2M3의 다리를 움직일 수 있는 공각이 생긴다. 그러면 M1M2M3를 움직여서 M16M17M18다리 옆에 가져다 놓음으로 앞으로 나가게 되는 것이다.



[그림 6] 모션 생성도구를 이용한 행위 데이터 생성

#### 다. 로봇 시뮬레이션

시뮬레이션 도구는 모션생성도구에서 만든 모션 행위데이터를 실제 환경과 유사한 가상의 환경에서 로봇의 행동을 수행하여 움직임이 자연스럽게 진행되는지를 점검하는 것이다. 모션 생성 도구는 로봇을 마우스를 이동시켜 모양을 만들기 때문에 물리엔진이 적



[그림 7] 시뮬레이터를 이용한 행위 데이터 수행

용되면 중력이 작용 돼서 물체가 계속 밑으로 내려가기 때문에 원하는 모양을 만들기 힘들다. 그래서 모션 생성도구와 시뮬레이션 도구를 분리시켜 동작하는 것이다. 도구는 분리되어 있지만 똑같은 데이터 처리기가 내장되어 있어 모션생성데이터가 시뮬레이션에서도 변환 작업 없이 그대로 사용될 수 있다. 그림 7은 모션 행위데이터를 수행할 수 있는 시뮬레이션 도구의 모습이다.

## 5. 분석

본 논문에서 제안한 로봇 개발 환경을 이용하여 개발하였을 때 얼마나 빨리 개발할 수 있는지를 비교하기 위해서 로봇 제어 경험이 없는 학생을 대상으로 1시간동안 교육을 하고 로봇의 움직임을 제어하도록 해 보았다. 총 3팀으로 나누고 팀당 5명으로 총 15명이 실험에 참가하였다. 각 팀은 과목을 수강하는 학생 중 프로그래밍 실력이 높은 1명, 중간 2명, 하위 2명으로 구성하였다. 로봇의 움직임은 앞으로 걷는 모습, 뒤로 걷는 모습, 옆으로 걷는 동작을 소프트웨어로 구현하고 실행하였다. 개발 정도는 요구사항, 설계, 구현, 수행을 단계별로 나누어 개발정도를 피 실험자가 체크하였다. 그리고 TTA의 지능형로봇 소프트웨어 시험평가 항목<sup>[11]</sup>를 사용하여 품질을 측정하였다. 측정값은 각점수를 100점으로 환산하여 합산하였다. 실험 집단 A군은 코드를 이용하여 개발하는 경우, B는 모션 편집기를 이용하여 개발 하는 경우, C는 본 논문에서 제안한 개발 방법을 사용한 경우로 구분하여 실행하였다. 단, 코드를 사용하는 경우 모터 제어 알고리즘을 제공하여 주었다. 모션 편집도구는 모터 값을 수치로 입력할 수 있는 도구를 말한다. 표 3은 로봇 개발 정도를 측정한 결과 이다. 요구사항, 설계, 구현, 수행의 단위시간은 1시간이고 품질평가는 100점 만점 이다.

그림 8은 각 그룹들의 수행시간을 비교한 결과이다. C그룹이 가장 빨리 개발한 것을 확인 할 수 있다. 이러한 결과를 통해 제안한 방법을 사용하면 로봇 개발을 쉽게 하여 로봇의 동작을 빠르게 생성 할 수 있고 다양한 응용동작을 만들어 낼 수 있다.

[표 3] 로봇 개발 정도 측정

구분		A군 (시간/점수)	B군 (시간/점수)	C군 (시간/점수)
앞으로 건기	요구사항	0.27	0.26	0.25
	설계	2.25	2.20	0.30
	구현	1.30	0.50	0.40
	수행	0.21	0.22	0.15
	품질평가	71	78	85
뒤로 건기	요구사항	0.23	0.24	0.29
	설계	1.5	1.40	0.28
	구현	1.32	0.45	0.38
	수행	0.21	0.20	0.12
	품질평가	75	80	87
옆으로 건기	요구사항	0.28	0.27	0.26
	설계	1.10	1.00	0.27
	구현	1.31	0.44	0.35
	수행	0.14	0.15	0.10
	품질평가	73	81	86

형태와 모양이 달라진다. 그리고 각각의 환경에 맞는 로봇을 만들기 위해서 프로토타입형 로봇을 제작하게 되면 별도의 비용과 시간이 추가로 소모된다. 또한 개발 후 문제점을 발견하더라도 수정이 어렵다. 본 논문에서는 자율탐색 로봇 설계를 위해 M&S(Modeling & Simulation) 환경을 제안하였다. 그리고 제안한 방법을 적용하여 6족로봇 개발하고 학생들을 대상으로 실험 군을 나누어 개발 수행시간을 비교해 보았다. 측정결과 제안한 M&S를 이용한 방법을 사용하였을 때 다른 그룹보다 5시간정도 빨리 개발한 것을 확인 할 수 있었다. 제안한 방법을 사용하면 다관절로봇 개발을 쉽고 빠르게 할 수 있다.

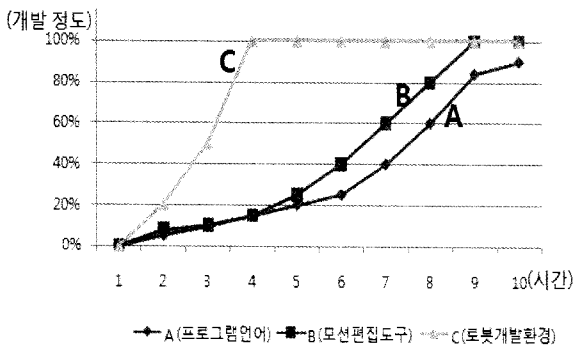
향후연구로 리팩토링 및 최적화 기술을 적용시켜 로봇시스템에 효율적인 코드를 자동 생성하는 방법을 연구 중이다. 또한 생성 모델 및 코드의 신뢰성 검증에 관한 연구가 진행중이다.

## 후 기

본 연구는 교육과학기술부와 한국산업기술재단의 지역혁신인력양성사업(2008~2009)으로 수행된 연구결과임

## 참 고 문 헌

- [1] 진태석, "미국의 국방로봇 최신통향", 주간기술동향, 통권 1312호, pp. 1~10, 2007. 9. 5.
- [2] W. Kim, R. Y. Kim, "Adapting Model Driven Architecture for Modeling Heterogeneous Embedded S/W Components", ICHIT2006, Vol. 2, pp. 705~711, 2006. 11.
- [3] Woo Yeol Kim, Hyun Seung Son, Young B. Park, Byung H. Park, C. R. Carlson, R. Young Chul Kim, "The Automatic MDA(Model Driven Architecture) Transformations for Heterogeneous Embedded Systems", SERP2008, USA, Vol. 2, pp. 409~414, 08. 7. 14.
- [4] Woo Yeol Kim, R. Young Chul Kim, "A Study



[그림 8] 로봇 개발의 수행시간 비교

## 6. 맺음말

자율탐색 로봇은 산악지형, 사막, 수중과 같은 다양한 환경 속에서 임무를 수행하며 지형에 따라 로봇의

- on Modeling Heterogeneous Embedded S/W Components based on Model Driven Architecture with Extended xUML”, KIPS Trans., Vol. 14-D, No. 1, pp. 83~88, 2007. 2.
- [5] H. Kim, Y. J. Cho and S. R. Oh, “CAMUS - A Middleware Supporting Context-aware Services for Network-based Robots”, IEEE Workshop on Advanced Robotics and Its Social Impacts, Nagoya, pp. 237~242, Japan, 2005.
- [6] OSGI, “The Dynamic Module System for Java”, <http://www.osgi.org/>
- [7] ER, “Evolution Robotics Software Platform (ERSP)”, <http://www.evolution.com/>
- [8] MSRS, <http://www.microsoft.com/robotics/>
- [9] 마이크로소프트 로보틱스 그룹, “Microsoft Robotics Studio(MSRS) 소개”, 2007. 6.
- [10] Open Dynamic Engine, <http://www.ode.org>
- [11] TTAS.KO-10.0266, “지능형로봇 소프트웨어 시험평가 항목”, <http://www.tta.or.kr>, 2007.