

제6권1호

2008년도

# 한국인터넷방송통신TV학회 춘계학술대회 논문집

일시 : 2008년 5월 31일(토)

장소 : COEX 본관 3층 311호

(서울시 강남구 삼성동 무역센터 코엑스)

홈페이지 : <http://www.iwit.or.kr>

주최 : (사)한국인터넷방송통신TV학회

동아방송예술대학 산학협력단

주관 : (사)한국인터넷방송통신TV학회



(사)한국인터넷방송통신TV학회

- ▶ 시간 : 14:30-16:00
- ▶ 장소 : COEX 본관 3동 311호

I 방송/통신/TV 분야(B) :

좌장 : 민경찬 소장 ((주) 한국기술연구소)

- B-1 ▶ 안전성 향상을 위한 비기능적 요구사항 기반 임베디드 S/W 시스템 개발 / 28  
[서진원, 김우열, 김동호, 김영철 (홍익대학교)]
- B-2 ▶ UHF TV 대역상의 Multimedia Application을 위한 CR Sensing MAC 테스트용 플랫폼 개발 / 32  
[황성호, 김기홍, 민준기, 이성수 (삼성전기중앙연구소)]
- B-3 ▶ 공중파 제거능력을 갖는 x-DSL 고속화 장치 개발 / 36  
[민경찬\*, 오호석\*\*, 강정진\*\*\* ((주) 한국기술연구소\*, KT미래기술연구소\*\*, 동서울대\*\*\*)]
- B-4 ▶ WPAN 모뎀용 전류모드 소자의 성능 개선에 관한 연구 / 41  
[김성권\*, 이광희\*, 조승일\*, 박천관\*, 차재상\*\* (목포해양대\*, 서울산업대학교\*\*)]
- B-5 ▶ 해상 데이터 통신을 위한 저전력 전류모드 신호처리에 관한 연구 / 45  
[김성권\*, 조승일\*, 조하나\*, 박천관\*, 차재상\*\* (목포해양대\*, 서울산업대학교\*\*)]



# 안전성 향상을 위한 비기능적 요구사항 기반 임베디드 S/W 시스템 개발

## Developing for the embedded software system based on safety of Non-functional requirements

서진원\*, 김우열, 김동호, 김영철

Jin-Won Seo\*, Woo-Yeol Kim, Dong-Ho Kim, R. Young Chul Kim

suh, john, ray, bob@selab.hongik.ac.kr

### 요 약

임베디드 시스템은 매우 정교하고 복잡한 개발 과정 때문에 개발 메커니즘의 체계화가 필요하다. 그중에도 안전성이라는 문제는 매우 심각하다. 임베디드 소프트웨어는 더욱 복잡성과 정교성 요구가 급증하는 추세이다. 본 논문은 소프트웨어 아키텍처 기반의 임베디드 시스템 개발에 있어서 비기능적인 면의 설계를 제안한다. 적용사례로 Movecar 로봇을 개발하여 비기능적인 안전성에 적용하였다. 아키텍처 단계에서 안전성을 고려한 설계를 통해 임베디드 시스템의 성능 향상을 확인하였다.

**키워드** : 소프트웨어 아키텍처, 비기능적 요구, 안전성, 임베디드 시스템

### I. 서 론

증가하는 사용자 요구를 만족시키고 시스템의 한정된 자원을 좀 더 효율적으로 사용하기 위해 소프트웨어 개발 메커니즘은 발전해왔다[1,7]. 임베디드 시스템[4]은 매우 정교하고 복잡한 개발 과정 때문에 개발 메커니즘의 체계화가 필요하다. 그중에도 안전성이라는 문제는 매우 심각하다. 임베디드 시스템이 탑재되는 우주왕복선의 시스템 오류의 사고나 항공기의 시스템 오작동 추락사고 등으로 사람의 생명을 위협하기도 한다.

이러한 문제점을 극복하기 위해 소프트웨어 아키텍처 메커니즘[2]을 임베디드 시스템을 적용하려는 노력이 있다. 시스템의 기능적 면도 중요하지만 신뢰성, 성능, 안전성, 모듈성 등과 같은 비기능적 요구사항을 통해 시스템을 향상시킬 수 있다고 본다. 이는 개발하고자 하는 타겟 시스템의 설계 목적(Design Goal)을 초점으로 개발 할 수 있기 때문이라고 본다.

본 논문은 소프트웨어 아키텍처 기반의 임베디드 시스템 설

계에 있어서 비기능적인 면의 설계를 제안한다. 적용사례로서 Movecar 로봇을 개발하여 비기능적인 요구사항 중 안전성을 적용하였다. 아키텍처 단계에서 안전성을 고려한 설계를 통해 임베디드 시스템의 성능 향상을 확인하였다.

논문의 구성은 다음과 같다. 2장에서는 관련된 소프트웨어 아키텍처들을 소개 하고 Polling 아키텍처의 비기능적 요구에 관하여 언급한다. 3장에서는 소프트웨어 아키텍처 기반의 임베디드 시스템의 적용사례를 들고 마지막으로 4장은 결론을 언급하였다.

### II. 관련 연구

#### 2.1 Polling 아키텍처

본 논문에서 정의하는 Polling 아키텍처는 컨트롤러가 끊임 없이 주변장치에 서비스가 필요한지 물어보는 소프트웨어 기법이다. 컨트롤러로 전송할 준비가 된 데이터를 가질 때, 주변 장치는 플래그를 설정하고 컨트롤러는 다음 Poll에서 인지한다. 어떤 플래그들이 설정되었는가에 따라, 컨트롤러가 다른 소프트웨어 루틴으로 점프하여, 이와 같은 다수의 주변장치들이

\* 홍익대학교 일반대학원 소프트웨어공학연구소실



연속적으로 Polling될 수 있다.

이러한 Polling 아키텍처의 각 주변장치를 지속적으로 Polling 해야 함으로 프로세서 자원의 낭비를 야기할 수 있고, 오버헤드가 발생할 수 있으며, 전송 데이터의 응답시간이 길어질 수 있다는 단점이 있다.

### 2.2 소프트웨어 아키텍처의 비기능적 요구

소프트웨어 아키텍처 기반의 임베디드 소프트웨어 시스템 개발에는 기본적으로 소프트웨어 개발 프로세스 기법 중 폭포수 모델 기반으로 적용하여 개발하였다.

개발단계에서 아키텍처 단계를 고려하여 비기능적 요구사항들에 따른 각 모듈마다의 체크리스트를 설정하였다. 체크리스트의 구성은 처음에는 유스케이스 이름을 설정하여 유스케이스를 식별하게 되며 관계중인 액터를 설정하여 이벤트에 영향을 주는 액터들을 나열하게 되어 액터들을 식별하여 정의한다. 이벤트의 흐름은 이벤트간의 사건을 나열하여 시스템 안에 있는 상호작용들을 구별할 수 있게 된다. 입구 조건에서는 처음에 이벤트가 시작하는 조건을 설명하여 정의하고 출구 조건에서는 시작하는 이벤트가 종료되어버리는 조건을 나타내도록 정의한다. 마지막으로 비기능적 요구사항에선 모듈들이 기능적인 행위 외에 다른 상태를 고려 할 수 있도록 나열하여 설명하여 정의한다[2].

## III. 비기능적 요구사항 기반 임베디드 S/W 개발

이번 장은 적용 사례인 Movecar 로봇의 하드웨어적인 측면과 소프트웨어적인 측면에 대하여 살펴보겠다.

### 3.1 Movecar 로봇

예제로 사용된 로봇(그림 1)은 Parallax사의 Javelin 모듈과 서보모터, 초음파센서를 조합한 것이다.

Movecar 로봇은 Java로 프로그래밍 할 수 있고 제어 프로그램을 시리얼포트를 통해서 대상 시스템의 플래쉬 메모리에 다운로드 하여 작동시킨다[5,6,7].



그림 1. Movecar 로봇

### 3.2 Movecar 로봇 시스템 개발

#### 3.2.1 요구 분석

시스템의 유스 케이스 다이어그램은 그림 2이다. 기능적 요구사항에서는 물체회피 유스케이스, 물체이동 유스케이스, 라인트레이스 유스케이스를 고려하여 설계하였다. 액터는 시스템으로 구현과 사용을 확장하여 이벤트의 흐름이 센서모듈로 전달하고 받은 이벤트는 제어모듈에서 모터모듈로 이동한다.

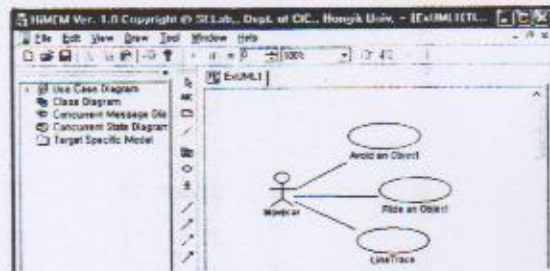


그림 2. HIMEM 1.0 도구를 통한 유스 케이스 다이어그램

#### 3.2.2 아키텍처

Movecar가 기본적으로 동작할 수 있는 기능적 요구는 앞으로 전진, 왼쪽 센서에 물체를 감지하면 우회전, 오른쪽 센서에 물체를 감지하면 좌회전 이다. 그리고 두 개의 센서가 동시에 감지했을 때 후진을 하고 난 뒤 좌회전 한다. 비기능적 요구 사항중 안전성을 고려하여 표 1과 같이 체크리스트를 작성하였다. 비기능적 요구는 각 유스케이스마다의 제약 조건들 중 물체 이동 유스케이스의 안전성은 Movecar위에 구슬을 올려놓으면 떨어지지 않도록 모터의 속도를 점진적으로 변화한다는 제약을 주어 비기능적 요구사항을 만족하는 유스케이스 체크리스트를 작성하였다.

표 1. 물체 이동 체크 리스트

| 유스케이스 이름  | 물체 이동  |  |
|-----------|--|--|
| 참여 액터     | Movecar  |  |
| 이벤트들의 흐름  | ① Movecar가 출발지점에서 출발한다.<br>② 정차지점에 정차하여 구슬을 올려 놓는다.<br>③ 구슬을 올려 놓은 상태로 이동한다.<br>④ 종착지점에 도착하여 정지한다. |  |
| 입구 조건     | 장애물이 초음파 센서에 감지 당했을 때  |  |
| 출구 조건     | 트랙을 한 바퀴 돌고 난 다음에는 완전히 정지한다.   |  |
| 비기능적 요구사항 | 안전성  | MoveCar위에 구슬을 올려놓으면 떨어지지 않도록 모터의 속도를 점진적으로 변화한다.       |
|           | 성능   | MoveCar의 동작 중이나 정지하고 있을 때 예도 sensor는 항상 감지를 하고 있어야 한다. |
|           | 정확성  | MoveCar의 회전동작은 처음과 나중의 사이 각을 약 30도를 유지해야 한다.           |

다음 설계 섹션에서는 현재 홍익대학교 소프트웨어 연구실에서 제안한 이중 임베디드 S/W 개발방법론 기반으로 개발한 HIMEM 1.0 도구를 사용하여 설계 및 구현에 대하여 알아본다[8,9,10].



### 3.2.3 설계

유스 케이스의 식별 과정 후 시스템을 제어클래스, 모터 클래스, 센서 클래스로 분할하였다. 제어클래스의 Forward() 메소드는 전진을 하면서 장애물을 만났을 때는 좌우에 있는 초음파센서가 감지하여 피하도록 Turn\_Left() 메소드나 Turn\_Right()를 수행하도록 하였다. 그림 3은 MoveCar에 대한 클래스 다이어그램으로서 제어클래스가 중심이 되어 모터 클래스, 센서클래스들을 관리하도록 연관관계로 정의하였다. 비기능적 요구사항이 반영된 체크리스트에서 안전성을 만족하기 위해서는 제어클래스의 forward() 메소드를 수정하였고 신뢰성을 만족하기 위해서는 On()메소드를 수정 하였다.



그림 3. HIMEM 1.0 도구를 통한 클래스 다이어그램

비기능적 요구의 안전성을 만족하기 위해서는 센서로 이벤트가 들어 왔을 때 전진을 천천히 하고 승차지점 이벤트와 정차지점 이벤트가 들어왔을 때 서서히 정지하는 상태를 유지해야 한다.

그림 4와 같은 시퀀스 다이어그램은 사용자와 시스템 혹은 객체와 시스템 간의 동적인 행위를 기술하며, 설계 단계에서 수행한다. 기능적 요구사항으로의 시퀀스 다이어그램으로는 장애물이 없을 때 센서객체에 메시지를 보내어 제어객체가 이를 판단하여 전진을 알리는 Forward메시지를 모터 객체에 보낸다. 물체를 만났을 때는 센서객체가 이를감지하여 제어객체로 메시지를 보내어 모터객체가 Move\_Right나 Move\_Left를 수행 할 수 있도록 메시지를 보낸다.

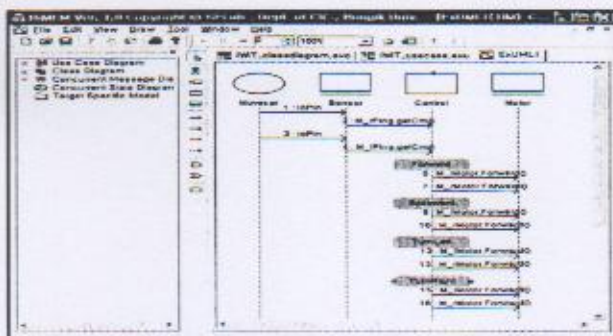


그림 4. HIMEM 1.0 도구를 통한 시퀀스 다이어그램

사용자가 MoveCar에게 메시지를 보냈을 때 비기능적 요구 사항을 만족하기 위해서는 MoveCar가 움직이는 동안 급작스런 속도를 내지 않도록 모터객체에게 메시지를 보내는 것이다. 또한 승차지점과 정차지점에 정확하게 정지할 수 있도록 제어하는 메시지를 모터객체에게 순차적으로 보내는 것이다.

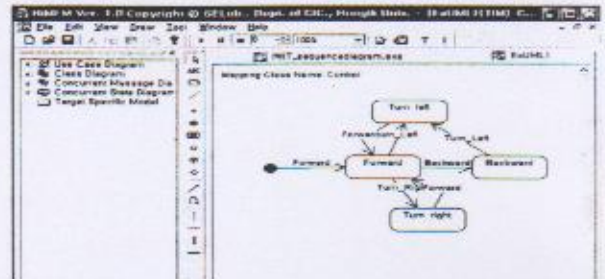


그림 5. HIMEM 1.0 도구를 통한 시퀀스 다이어그램

상태 다이어그램은 하나의 동적 객체의 상태 및 상태 변화를 기술한 다이어그램이며, 소프트웨어 개발 프로세스 중 설계 단계에서 수행한다. 본 시스템의 상태 다이어그램은 그림 5와 같다[7]. 기능적 요구의 상태 다이어그램으로서 는 센서로부터 이벤트가 들어 왔을 때 전진, 좌회전, 우회전, 후진을 만족하는 행위를 하게 되며 비기능적 요구의 안전성과 신뢰성을 만족하는 행위로서는 센서로 이벤트가 들어 왔을 때 전진을 천천히 하고 승차지점 이벤트와 정차지점 이벤트가 들어왔을 때 서서히 정지하는 것이다.

### 3.2.4 구현

기능적 요구사항의 구현은 제어클래스의 Forward(), Backward(), Turn\_Right(), Turn\_Left() 메서드를 작성하는 것이다. 전진과 왼쪽에 장애물이 있을 때는 우회하고, 오른쪽에 장애물이 있을 때는 왼쪽으로 비껴가고 정면에 장애물이 있을 때는 후진을 한 뒤 우회하여 장애물을 피하는 방법을 코드로 작성하였다.

본 논문에서 수행된 비기능적 요구사항의 요건으로서는 안전성이다. 가장 기본적인 아키텍처인 Polling 방식을 채택하여 프로그래밍을 하였다. 시스템에서 구슬을 사람으로 간주하여 구슬이 떨어지면 안전성을 만족하지 못하는 실험이다. 안전성을 만족하기 위해서는 속도의 변화가 점진적으로 증가하거나 감소하고 주행 중에는 일정한 속도를 유지하는 것이 중요하다. 그 과정 중에서도 속도의 급변화가 이루어지는 부분은 바로 처음 출발할 때와 정지할 때의 속도이다. 정지해 있다가 출발을 할 때 순간적인 펄스를 통해 급출발을 하게 되는데 이것을 제어하기 위해서는 속도를 서서히 올릴 수 있게 해야 한다. 그리고 주행 중의 로봇을 정지시킬 때에는 급제동을 하게 되는데 이것을 제어하기 위해서는 속도를 서서히 줄



이는 방법을 채택해야 한다.

```

public void Forward()
{
    if(m_cnPosition == '1')
    {
        CPU.pulseOut(STOP - SPEED,m_nPn);
        switch(N_nForwardCount)
        {
            case 1:
                Turn_left.Setspeed(5);
                Turn_right.Setspeed(5);
                break;
            case 20:
                Turn_left.Setspeed(10);
                Turn_right.Setspeed(10);
                break;
            case 40:
                Turn_left.Setspeed(15);
                Turn_right.Setspeed(15);
                break;
            case 60:
                Turn_left.Setspeed(20);
                Turn_right.Setspeed(20);
                break;
            case 80:
                Turn_left.Setspeed(25);
                Turn_right.Setspeed(25);
                break;
            case 100:
                Turn_left.Setspeed(30);
                Turn_right.Setspeed(30);
                break;
        }
    }
}
    
```

그림 6. 제어 클래스의 비기능적인 안전성

비기능적 요구사항의 안전성을 높이기 위해 그림 6의 빨간 상자 부분과 같이 변경하였다. 제어클래스 안에 있는 Forward()메소드를 스위치 케이스(switch-case)문을 사용하여 6단계의 속도로 구분하게 하여 점진적으로 속도가 증가하도록 하였다.

#### IV. 결론

임베디드 시스템을 소프트웨어 아키텍처관점에서 기능적 과 비기능적 요구사항으로 구분하였고 안전성을 고려할 수 있도록 설계 및 개발하였다. 시스템에 사용된 Movecar 로봇은 Polling 메커니즘으로 구현하여 간단하게 연산을 수행할 수 있었다. 처음 시스템은 단순하게 기능적인 면만을 고려하여 설계, 구현하여 구슬이 트랙을 돌다가 떨어지고 정확히 정착할 지점에 정착하지를 못하였다. 비기능적 요구 중 안전성을 높이기 위해 소프트웨어를 여러 번 수정하였다. 기능적 위

주의 시스템은 다른 여러 상황들을 고려하기가 매우 어렵다. 기능 위주로만 고려해서는 결코 좋은 시스템을 만들 수가 없기 때문에 다양한 측면을 고려하여 임베디드 시스템을 설계해야 한다.

향후 과제로 비기능적인 요구사항인 안전성뿐만 아니라 신뢰성, 성능, 구현성 등 다양한 요소들을 사용하여 사용자 요구를 만족할 수 있도록 임베디드 시스템에 적용할 것이다.

본 연구(2008년)는 교육과학기술부와 한국산업기술평가관리원의 지역혁신인력양성사업으로 수행된 연구 결과임.

#### 참고 문헌

- [1] Bernd Bruegge, Allen H.Dutoit "International edition Object-Oriented Software Engineering : Using UML, Patterns, and Java", pp.3-9 .
- [2] Mary Shaw, David Garlan "Software Architecture, Perspectives on an Emerging Discipline", Prentice Hall, pp.43-51
- [3] Avi Silberschatz, Peter Baer Galvin, Greg Gagne, 'Operating System Concepts', John Wiley & Sons, 2004.
- [4] Qing Li, 'Real-Time Concepts for Embedded Systems', CMP, 2003.
- [5] PARALLAX.INC , "Robotics! Student workbook version 1.5", pp.42
- [6] 최재현, 김영철, 김경창, "Polling 메커니즘 기반의 임베디드 소프트웨어 시스템 개발", 모바일학회 추계학술대회 pp.245-250 2006
- [7] 서진원, 오영덕, 김영철, "소프트웨어 아키텍처 기반의 임베디드 시스템 개발", 모바일학회 추계학술대회 pp.240-244 2006
- [8] 손현승, 김우열, 서윤숙, 김동호, 김동우, 김계수, 김영철 "이종 임베디드 소프트웨어를 위한 코드 생성 메커니즘 및 지원도구", 한국소프트웨어공학회, 170-177,2007.08.24
- [9] 김예진, 손현승, 김우열, 서진원, 김동호, 서윤숙, 류동국, 김영철, "확장된 xUML을 사용한 장애물 회피용 소형 무인차 모델링 연구", 한국소프트웨어공학회,108-116, 2007.08.24
- [10] Woo Yeol Kim, R. Young Chul Kim, "A Study on Modeling Heterogeneous Embedded S/W Components based on Model Driven Architecture with Extended xUML", The KIPS Trans. Vol. 14-D, No. 1, 2007.02