

# 로봇 소프트웨어 프레임워크 동향과 발전방향

## A Review and Outlook of Robotic Software Frameworks

최 병 옥<sup>1</sup>

Byoung Wook Choi<sup>1</sup>

**Abstract** Robotic software has been dramatically complicated as performing intelligent service tasks. These types of robots demand a very powerful software framework to make them easy. Robotic software framework means an integrated software environment that simplifies jobs of robotic software engineer by providing tools, reusable components, and runtime environments. Finally it reduces the project cost. There are lots of works related with them. Among them we focus on five frameworks that are MSRDS, ERSP, OROCOS, OpenRTM, and OPRoS. In terms of intelligent service robot, the study on robotic software frameworks is very important. And outlook on them is also very important in the sense of that the robotic software frameworks should be used to initiate service robot market.

**Keywords:** Robot Software framework, MSRDS, ERSP, OROCOS, OpenRTM, OPRoS

### 1. 서론

소프트웨어에 있어서 프레임워크(framework)는 그 의미가 다양하다. 그러나 일반적으로 소프트웨어 공학에서 말하는 프레임워크 기술은 특별한 기능을 제공하는 사용자 프로그램에 의하여 선택적으로 기능이 보장되거나 특정화가 가능한 소프트웨어 기능의 추상화된 개념이다<sup>1)</sup>. 프레임워크는 일반적으로 API(Application programming interface)로 잘 정의되어서 재사용이 가능한 소프트웨어라는 점에서 소프트웨어 라이브러리와 유사하나 전체적인 프로그램의 제어 흐름이 사용자에게 의하여 지배되는 것이 아니라 프레임워크에 의하여 결정된다는 점에서 큰 차이를 가지고 있다. 잘 알려진 바와 같이 자바의 JDK 및 JRE 그리고 마이크로소프트의 .NET framework가 프레임워크의 예라고 볼 수 있다. 또한, 일반적으로 프레임워크와 개발환경이 통합되어 있는 것을 플랫폼(Platform)이라고 부르고 있다.

프레임워크 정의로 비추어 볼 때 먼저 로봇 산업의 현실을 살펴보는 것이 기능 결정에 도움이 될 수 있다. 일반적으로 산업 현장에서 자동화 프로젝트를 수행할 경우 약 80%에 해당하는 작업이 소프트웨어 개발과 상용화를 포함하는 시스템 통합 작업이다. 따라서 로봇 소프트웨어

개발자에게 소프트웨어 개발과 상용화를 포함한 시스템 통합 작업을 간단하게 하여주고 결과적으로 프로젝트의 개발 시간을 줄이는 신뢰성 있는 기능의 지원이 필요하다. 이에 대한 해결점이 로봇 소프트웨어 프레임워크이다.

요구 사항으로는 추상화된 아키텍처를 가지고 있어서 사용자에게 로봇 응용과 관련하여 유연한 구조를 제공하여야 한다. 또한 프레임워크에서 시스템의 제어 흐름을 관장하고 사용자는 주어진 제어 흐름 내에서 필요한 작업을 수행하여 단순한 기능적 오류가 시스템의 오류로 발전하지 말아야 한다. 또한 관련된 기능과 데이터가 인캡슐화(encapsulation)된 컴포넌트 형태로 재 사용이 가능한 추상화된 소프트웨어 기능적 모듈이어야 하며 기능이 표준화되고 안정화된 구조이어야 한다. 그리고, 로봇 소프트웨어 프레임워크에서 개발된 비즈니스 컴포넌트 자산이 재 사용이 되고 궁극적으로 비즈니스 컴포넌트와 로봇 애플리케이션 프레임워크가 결합하여 상위 수준의 비즈니스 프레임워크로 발전할 수 있어야 한다.

현재 로봇 소프트웨어 프레임워크는 목적에 따라 대단히 많이 연구되고 있다<sup>2)</sup>. 이 중에서 비교적 많이 알려진 MSRDS(Microsoft Robotics Developer Studio)<sup>3)</sup>, ERSP(Evolution Robotics Software Platform)<sup>4)</sup>, OROCOS(Open ROBOT Control Software)<sup>5)</sup>, OpenRTM-aist<sup>6)</sup>와 OPRoS(Open Platform for Robotic Services)<sup>7)</sup>에 대하여 특징을 중심으로 기능을 살펴보기로 한다.

주요한 기능을 비교해 봄으로써 국내 연구자에게 필요한 기술적 배경을 제공하는 것을 목적으로 하고 있으며, 로봇 소프트웨어 프레임워크가 궁극적으로 로봇 산업의 활력소와 새로운 기회 창출을 위하여 어떠한 목표를 가지고 진행되어야 하는 점을 기술하고자 한다. 이와 같은 목적을 위하여 현재 이동통신 업계에서 많이 언급되고 있는 프레임워크와 이동통신 산업 그리고 소프트웨어 응용 프로그램 개발자의 상생의 생태계를 비교하여 로봇 소프트웨어 플랫폼의 발전 방향을 제시하는 것을 목적으로 한다.

## 2. 로봇 소프트웨어 프레임워크 기능 검토

이 장에서는 로봇 소프트웨어 프레임워크의 특징과 기능을 분석하는데 동일한 판단 기준으로 비교하는 연구는 무리가 있다. 왜냐하면, 각각이 지향하는 목적이 다르고 컴포넌트로 구성된 시스템의 구조인 아키텍처도 서로 상이함에 따라 기술적인 기능 보다는 결과론적인 응용에 대한 비교가 적절할 것이다. 따라서, 본 논문에서도 이와 같은 관점에서 연구를 진행하였다.

### 2.1 MSRDS

MSRDS는 2006년에 처음으로 소프트웨어를 공개한 이후 2008년 새로운 버전으로 현재 약 20만 번 이상의 다운로드와 50개 이상의 파트너를 구축하고 있다. MSRDS는 로봇 개발자를 위한 개발 플랫폼으로서 다양한 계층의 사용자와 하드웨어 그리고 응용 시나리오들을 지원하고 있다<sup>8)</sup>.

MSRDS는 현재의 로봇 개발 과정에서 겪고 있는 이식성 문제와 재사용성 문제를 해결하고, 로봇 종사자들에게 보다 쉬운 접근성을 제공하기 위해 만들어진 개발 도구로서, PC가 하드웨어 수준에서 기업 및 개인들에게 관심을 끌던 초창기에 다양한 하드웨어에 대한 이식성을 제공하기 위해 만들어진 Microsoft의 BASIC 개발 틀이 개인용 컴퓨터 시대를 이끌고 정보화 시대를 이끌었듯이 MSRDS를 통하여 향후 지능화된 로봇 시대를 이끌어 낼 수 있을 것으로 기대하면서 개발되었다.

그림 1.의 기능에서 보듯이 기본적으로 Windows/Windows CE 계열의 운영체제 상에서 .NET 프레임워크를 기반으로 하여 동작하는 소프트웨어 플랫폼이다. 또한 기능적으로는 DSS(Decentralized System Service)와 CCR(Concurrency and Coordination Runtime)의 실행환경을 기반으로 한다. 사용자 편의를 위하여 VPL(Visual Programming Language)을 지원하고 있다. 그리고 소프트웨어 구성에서는 서비스 기반의 아키텍처를 가지고 있으면 포트를 통하여 서비스 간의 통신이 이루어지는 전형적인

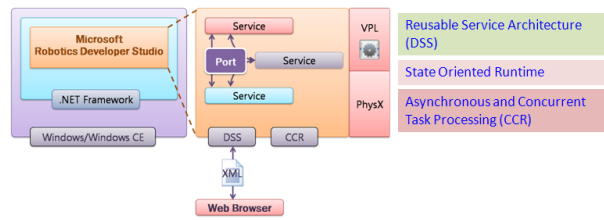


그림 1. MSRDS 아키텍처 및 주요기능

서비스 기반 소프트웨어 아키텍처를 기반으로 하고 있다. 따라서 개발자는 간단한 프로그램의 경우 VPL을 이용하여 응용 프로그램의 작성이 가능하지만, 전문적인 프로그램을 위하여서는 .NET 프레임워크 환경에서의 C# 프로그램에 익숙하여야 하며 서비스 기반의 소프트웨어 구조를 이해하여야 프로그램 작성이 용이하다<sup>9)</sup>.

주요 특징을 요약하면 다음과 같다.

- 서비스 기반의 런타임 아키텍처
  - 로봇 S/W의 이식성과 재사용을 제공
  - 서비스 인터페이스와 해당 서비스가 제공하는 기능에만 초점
- 분산 애플리케이션 패턴 지원
  - 여러 대의 컴퓨터 실행 환경 지원
  - 구조화된 메시지 처리와 이벤트 통지
- 동시처리 및 제어기술
  - .NET 프레임워크 활용
  - 단순한 코딩으로 동시성 문제 해결
- 시뮬레이션 환경
  - 하드웨어와 유사한 시뮬레이션 환경 제공
  - 개발 기간 단축 및 코드의 생산성 향상
- 비주얼 개발 언어 지원
  - 쉬운 로봇 S/W 프로그래밍 지원

MSRDS의 시뮬레이션 환경은 Ageia사의 PhysX 엔진을 이용하며, Microsoft XNA 프레임워크를 기반으로 구성되었다. 시뮬레이터는 하드웨어 개발 이전에 시뮬레이션 로봇을 여러 개발자가 동시에 활용함으로써 개발 기간을 단축하고, 코드의 생산성과 품질 향상에 이용 가능하다. 또한 물질세계에서의 중요한 요소를 모두 구현 할 수 있어서 그림 2와 같이 기존의 이동 로봇 위주의 시뮬레이션이 다양한 응용 분야로 발전하고 있다.

정책적으로는 MSRDS 기반 소프트웨어를 로봇에 내장하기 보다는 PC의 컴퓨팅 파워와 로봇의 이동성을 결합한 구조에 치중하고 있으며 대학교 중심으로 사용자 확대하고 있다. 따라서, 3~4년 후이나 로봇에 내장된 MSRDS 응용 분야를 목표로 하고 있다.

이에 비하여 MSRDS 기반 소프트웨어는 윈도우 계열

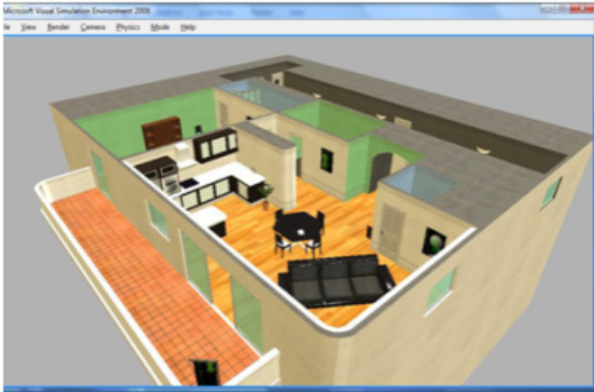


그림 2. 시뮬레이션 예제

운영체제를 필요로 한다는 점과 MSRDS는 무료이나 개발을 위한 도구는 유료라는 단점이 있다. 그리고 하드웨어에 내장하여 동작할 경우에는 무거운 구조의 프레임워크가 필요하며 별도의 런타임 비용을 지불하여야 하는 단점이 있다. 또한 로봇의 기능을 구현하는 도구 및 환경과 개발 표준이 없으며, 현재 상태에서는 로봇 업체의 요구를 빠른 시간 내에 수용하지 못하고 일반 소프트웨어 개발자 위주의 확대에 주력하고 있는 실정이다.

## 2.2 ERSP

2002년 이동 로봇을 개발하기 위한 소프트웨어 플랫폼으로 출시된 Evolution Robotics사의 ERSP는 다양한 로봇 시스템에 대한 소프트웨어를 개발하거나 프로토타입을 제공해주는 소프트웨어 개발 도구이다<sup>4)</sup>. ERSP의 유연한 소프트웨어 구조는 다양한 드라이버와 작업과 같은 소프트웨어 모듈을 포함하고 있어서 쉽고 빠르게 진보적이고 강인한 로봇 시스템 개발을 가능하도록 한다.

ERSP는 그림 3과 같이 계층적 구조의 로봇 소프트웨어 아키텍처를 채택하고 있으며, 시각적 물체인식 모듈(ViPR) 시각 기반 자율주행 모듈(vSLAM), 로봇 소프트웨어 아키텍처(ERSA)와 상호 작용 모듈(Interaction)의 4개의 모듈이 C++ 라이브러리로서 서로 간에 독립되어 있으며, 로봇의 행동을 제어함에 있어서 강력한 아키텍처를 제공하고 있다<sup>4)</sup>.

그림 3에 나타낸 ERSP 비전 모듈인 ViPR 물체인식 모듈은 물체의 확립된 특징점 추출 SIFT(Scalable Invariant Feature Transform) 알고리즘으로서, 독자적으로 개발한 Matching 알고리즘과 함께 일체화시킨, 물체인식 소프트웨어로서 C++ 라이브러리이다. 이는 SONY의 제 3세대 AIBO, NTT DoCoMo의 N902iS 시리즈의 휴대전화에 물체인식 소프트로 활용되며 SKTelecom의 PictureX 소프트웨어에 응용되고 있다.

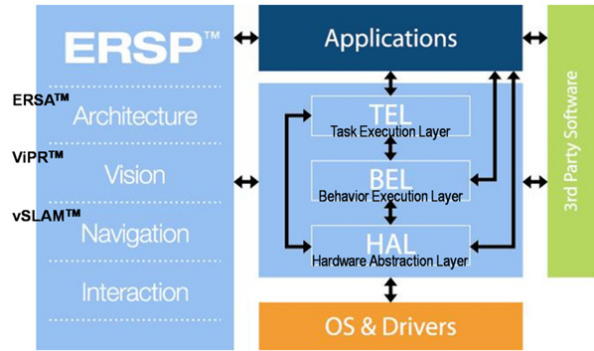


그림 3. ERSP 전체 구성

vSLAM(visual Simultaneous Localization And Mapping) 시스템은 2가지의 형태로 Front-End는 랜드마크의 생성과 인식을 Back-End는 로봇의 위치추정과 지도의 업데이트를 관리한다. 일반적으로 지도작성과 위치확인에 비전을 이용하며, 탐사, 경로계획 및 장애물 회피 그리고 점유 그리드 지도 작성이 가능하다. 또한, 환경변화에 강한 적응력이 있으며, 예상치 못한 상태에서도 급격히 성능이 떨어지지 않는 양도 구성되어 있다.

그리고 Interaction 모듈은 HRI(Human robot interaction) 개발을 위한 다양한 컴포넌트를 제공한다. 명령 수행을 위한 인터페이스와 로봇이나 디바이스 제어를 수행할 수 있다. 현재 구성된 기술은 음성 인식과 감정과 개성 표현 모듈 그리고 제스처 인식 모듈이 제공된다.

그림 4는 ERSP Architecture에 포함된 계층 구조와 컴포넌트 구성을 나타낸다.

- TEL(Task Execution Layer) 계층
  - 태스크의 복잡한 실행 조정
  - BEL과 HAL 계층에 대한 인터페이스 제공
  - 태스크는 연속적 또는 병렬적 실행이 가능
  - 태스크 실행은 사용자가 지정한 이벤트에 의하여 비동기적으로 기동
- BEL(Behavior execution Layer) 계층
  - 피드백 그룹의 태스크를 지원하는 기능
  - 태스크 개발을 위한 컴포넌트 제공
  - XML 파일을 통한 행위 특성 정의
- HAL(Hardware Abstraction Layer) 계층
  - 운영체제 추상화 및 이식성을 보장
  - 하드웨어 추상화 제공

ERSP의 계층 구조에서 특징적인 BEL 과 TEL을 구분하면, TEL 계층은 인간의 반사 신경에 해당하여 센서 등으로부터 정보를 얻어 로봇을 제어하면서 피드백 루프를 실현함으로써 동기화된 작업과 데이터 구동 작업을 수행한다. TEL 계층은 인간의 뇌에 해당하여 여러 가지의 타이

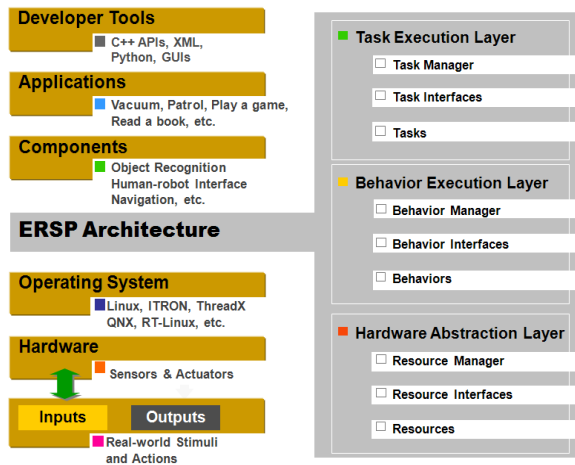


그림 4. ERSAP 아키텍처 및 주요 컴포넌트

밍으로 발생하는 상황 및 변화에 대하여 유연히 대응하여야 하는 경우에 사용하는 것으로 비동기 작업과 이벤트 구동 작업에 사용되어 진다.

ERSAP는 태스크의 순차적, 비동기적 실행을 지원하며 계층적 모듈화된 구조로 사용이 편리하다. 또한, 소니 AIBO, Bandai, ZMP 및 유진 로봇 등 성공사례를 가지고 있으며, 물체 인식 모듈의 경우 로봇 업계 이외에서도 많이 사용되고 있다. 윈도우 및 리눅스를 지원하며 높은 유연성 및 재사용 특징을 가지고 있으며 비교적 간단한 저가의 장비(카메라 1개, 엔코더)로 복잡한 공간에 대한 맵 생성을 할 수 있는 고성능 알고리즘이다.

위와 같은 장점에 비하여 통신 프로토콜 및 콘텐츠에 대한 지원이 없으며, 시뮬레이터 등과 같은 체계적 개발 도구가 지원되지 않음에 따라 주어진 모듈의 사용은 용이하나 확장된 개발이 쉽지 않다. 따라서 새로운 컴포넌트를 개발하기가 쉽지 않으며, 숙련된 개발자가 필요하다. 국내에서 많이 연구되고 있는 바와 같이 정보통신기술과 네트워크 인프라를 이용한 로봇 산업의 발전에 대한 통합 전략이 없으며, 서버와의 접속을 이용하여 기능을 확장하는 네트워크 로봇의 개념으로 확장이 어렵다. 그리고 ERSAP 아키텍처에 의존적인 특징이 있다.

### 2.3 OROCOS

OROCOS(Open ROBOT COntrol Software)는 2000년 12월에 로봇 제어 소프트웨어 개발을 위한 오픈 소스 프로젝트로 시작되어 EU (유럽연합)의 EURON(European Robotics Network)과 결합하여 벨기에의 K.U.Leuven, 프랑스의 LAAS Toulouse, 그리고 스웨덴의 KTH Stockholm을 주된 파트너로 하여 2001년 시작되어 3년의 프로젝트로 2002년부터 결과를 발표하고 있다<sup>[3]</sup>.

OROCOS는 현재 그림 5와 같이 4개의 주요한 라이브러리로 구성되어 있는데, RTT(Real-Time Toolkit), KDL(Kinematics and Dynamics Library), BFL(Bayesian Filtering Library) 및 OCL(OROCOS Component Library)을 통하여 다한 구현 모델을 제공 중이며, 다양한 컴포넌트 기반의 프로그래밍 인터페이스의 개발하는 것을 목적으로 한다.

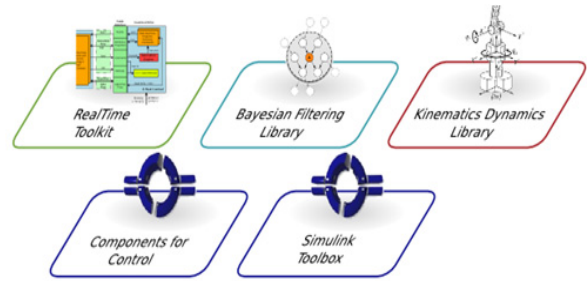


그림 5. OROCOS 소프트웨어 구성

각각의 소프트웨어의 특징을 보면 RTT는 자체가 응용 프로그램은 아니며 C++ 기반으로 로봇 관련 응용 프로그램을 작성할 때 함수와 도구를 제공한다. 특히 실시간성 지원과 컴포넌트 기반 응용으로 컴포넌트 간 통신과 배포에 대한 API와 XML 구성을 제공한다. KDL은 실시간으로 기구학 및 동역학 모델링 및 연산을 하기 위한 C++ 라이브러리로서 독립적인 프레임워크를 구성하고 있다. BFL은 동적 베이저안 네트워크(Dynamic Bayesian Networks)에서 추론을 위한 독립적인 프레임워크를 제공하여 로봇 응용에서 많이 사용되는 확장 칼만 필터(Extended Kalman Filters) 및 파티클 필터(Particle Filters) 등 반복적 정보 처리 예측 알고리즘을 제공한다. OCL은 쉬운 제어 컴포넌트 사용을 위한 컴포넌트를 제공하는 것으로 컴포넌트 관리와 제어 및 하드웨어 접근 처리를 위한 컴포넌트를 동시에 제공하고 있다.

소프트웨어 컴포넌트는 그림 6과 같이 미리 정의된 컴포넌트를 이용하며, RTT를 이용하여 자신의 컴포넌트를 구현하고 컴포넌트들의 연결함으로 제어 응용 프로그램을 구성한다. 제어 컴포넌트는 잘 정의된 구조로 Properties, Events, Methods, Commands, Dataflow의 인터페이스를 구현하고 있다. 또한 응용 템플릿은 잘 동작하는 컴포넌트 집합으로 경로 계획, 위치 제어, 하드웨어 처리 및 데이터 보고와 같은 것으로 구성되어 있어서 사용자가 자신의 응용 프로그램 작성에 활용할 수 있게 구성되어 있다.

가장 큰 장점은 실시간성 보장에 관한 연구를 중요한 하나의 과제로서 다루고 있는 점이다. 또한 사용 가능한



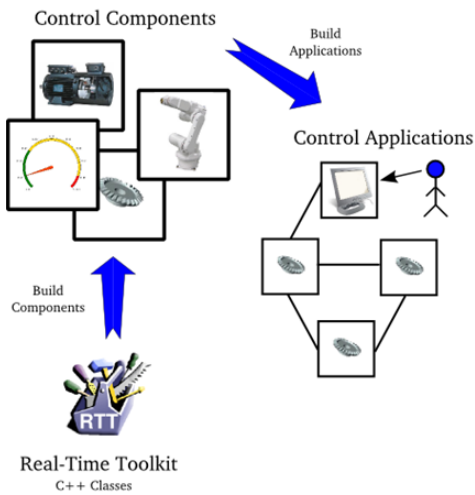


그림 6. ORCOS 기반 제어 응용

라이브러리 구축이 프로젝트의 중요한 목적으로 진행되고 있으며, ORCA(Open Robot Controller Architecture) 등 컴포넌트 기반의 로봇 S/W 프로젝트로 발전하고 있다<sup>[10,11]</sup>. 기술적으로는 다양한 미들웨어 옵션을 이용할 수 있으며, 다양한 컴포넌트 인터페이스로 구현이 가능하도록 정의되어서 오픈 소스 프로젝트와의 연계를 통한 컴포넌트 기반 라이브러리를 제공하여 사용자의 편의를 높이고 있다.

부족한 구성으로 지적할 수 있는 점은 프로젝트가 로봇 소프트웨어 컴포넌트 인터페이스에 주력하면서 제어 구조는 관여하지 않고 있다. 따라서, 컴포넌트 단위의 기능 설계와 컴포넌트의 조합을 통해 로봇 제어를 구현함으로써 계층적 구조의 시스템 구성이 쉽지 않다. 또한 오픈 프로젝트의 성격상 전체를 총괄하는 기술 지원이 안되고 있으며, 계속된 버전 증가에 따른 기술의 발전도 사용자가 사용하는데 부담으로 작용하고 있다. 그리고 개발자에게 중요한 단계별 개발 도구가 존재하지 않고 있으며, 지능형 서비스 로봇으로 발전을 위하여 필요한 로봇과 원격 서버간의 멀티미디어 접속 개념이 존재하지 않는다.

### 2.4 RT 미들웨어

RT 미들웨어는 신에너지 산업기술 종합개발기구 주관으로 2002년부터 시작되어 진행되었다. 로봇으로부터 로봇 기술인 RT(Robot Technology)로의 목적으로 진행되었으며, 현재는 일본 산업기술 종합연구소인 AIST(Advanced Industrial Science and Technology)를 중심으로 RTM(RT-middleware)를 개발하고 있다<sup>[6]</sup>.

RT 기술은 그림 7과 같이 기존의 로봇 개발에서 반복적으로 이루어진 소프트웨어 작업을 분산 객체 미들웨어

CORBA(Common Object Request Broker Architecture) 기반의 분산 네트워크를 통하여 표준 컴포넌트를 통하여 로봇 소프트웨어를 구현하는 것이다. 또한 국제적으로 OMG(Object Management Group)의 표준 기능의 표준화에 노력하고 있으며, 구현 방법론을 구축하여 개방형 로봇 시스템 아키텍처의 보급 및 실증을 목표로 하고 있다. 현재 AIST를 중심으로 실증 시스템 구축을 위한 로봇 시스템을 개발하고 있다<sup>[12]</sup>.

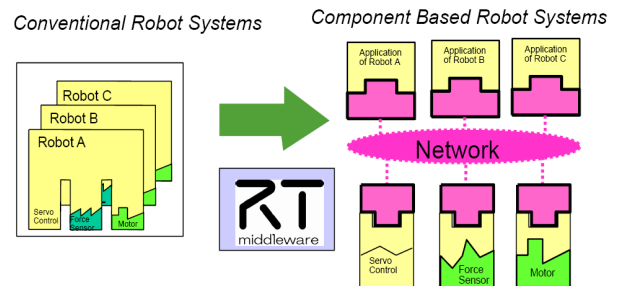


그림 7. RT 기술의 개요

RTC(RT Component)는 표준화된 인터페이스를 통하여 디바이스 제어, 알고리즘과 응용과 같은 논리를 구현한 컴포넌트를 의미하며, RTM은 네트워크를 통하여 분산화가 가능한 RTC가 동작하는 실행 환경이다. 따라서, 로봇 요소인 RTC를 분산 통신 네트워크를 통해 조합함으로써 다양한 로봇의 구축을 가능하게 하는 RT 미들웨어 기술을 의미한다.

그림 8은 OMG를 통한 표준화와 RTM의 개념을 나타내고 있다<sup>[13]</sup>. 일반적으로 OMG에서는 개념적인 PIM(Platform Independent Model)과 구현이 가능한 기술을 포함한 PSM(Platform Specific Model)을 정의한다. 그러나 이와 같은 정의를 이용하여 소프트웨어를 실장하기 위하여서는 각자 인터페이스나 사양의 추가 확장이 필요하다. 이와 같은 개념이 설계 모델이 되며, 하드웨어에 실장되어 동작하기 위하여 RTM의 개념이 필요하다. 설계 모델을 실장하는 미들웨어는 다양하게 구현이 가능하며 AIST에서 가장 활발히 개발하여 발표한 미들웨어는 OpenRTM-aist 모듈이다. 따라서, 각 로봇은 운영체제와 RTM 그리고 RTC로 계층적으로 구성되며, 각각의 서로 다른 운영체제를 기반으로 구성된 각 로봇의 RTC는 RTM을 통하여 동적으로 결합되고 통신이 가능하게 된다.

로봇 소프트웨어 표준을 위하여 OMG에서 Robotics Working Group을 형성하여 일본의 AIST와 미국의 RTI의 제안을 채택하여 로봇 내부 소프트웨어의 컴포넌트 규격인 RTC를 표준으로 채택하였다. 이와 같은 국제 표준

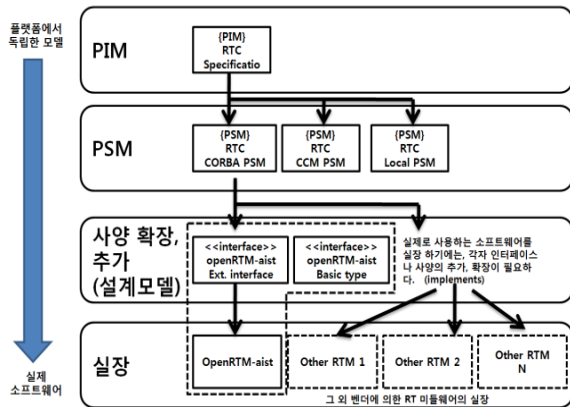


그림 8. 표준화 모델과 RTM

을 구현한 구현 모델이라는 점이 가장 큰 장점이다. 또한 AIST를 중심으로 도구와 구현 모델의 소프트웨어가 개발되어 배포되고 있다<sup>[12]</sup>. 기본적으로 컴포넌트 규격으로 인하여 소프트웨어 재사용을 목표로 하고 있으며, 실시간 제어를 가능하게 한 인터페이스 방법도 제안되어서 다양한 로봇 응용이 가능하다.

그러나 RTM은 RTC의 실행 환경으로 로봇 내부 소프트웨어의 컴포넌트 규격을 지원하고 있다. 따라서 지능형 서비스 로봇과 네트워크 로봇의 개념을 위한 컴포넌트 소프트웨어의 배포와 실행에 대한 지원이 없다. 또한 멀티미디어 지원을 위한 컴포넌트 지원이 아직 이루어지고 있지 않으며, 지원 도구도 부족한 실정이다. 또한 실행 태스크와 제어 구조에 대한 모델을 제공하지 않고 있어서 RTC를 이용한 제어 관점의 모델은 사용자가 독자적으로 구현하여야 한다.

### 2.5 OPRoS 프로젝트

OPRoS(Open Platform for Robotic Services)는 기존에 정보통신부에 의하여 2004년부터 진행된 RUPI(Robot Unified Platform Initiative)와 산업자원부 중심의 전략기술개발시범사업인 SPIRE(S/W Platform Initiative for Robotics Engineering) 개발이 정부 통합과 함께 통합되어 2008년부터 지식경제부 지원으로 개발 중인 로봇 소프트웨어 프레임워크이다.

목적은 네트워크 기반 로봇을 위하여 표준화되고 재사용이 가능한 개방형 소프트웨어 프레임워크이며, 2009년 10월부터 소스를 공개하고 있다<sup>[7]</sup>.

구성을 보면 그림 9와 같이 로봇 내부 소프트웨어와 네트워크 로봇을 위한 컴포넌트 컨텐츠 서버로 실행 환경을 구성하고 있다. 로봇 내부 소프트웨어를 위하여서는 컴포넌트 기반으로 재사용을 목적으로 하여 구성되었으

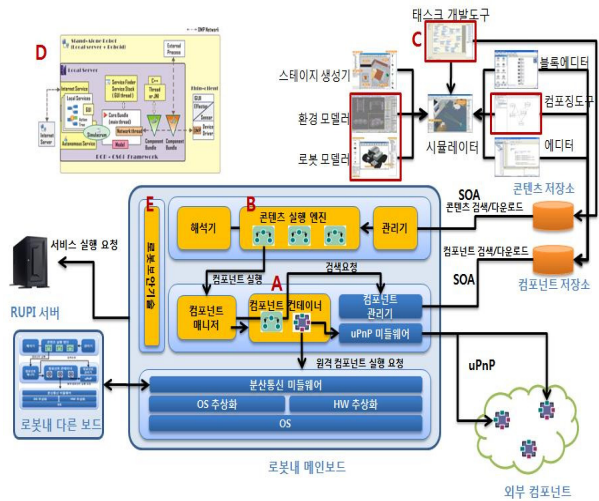


그림 9. OPRoS 구성

며 서버와의 협업과 소프트웨어 배포를 위한 관리자가 구성되어 있다. 그리고 컴포넌트 컨텐츠 서버는 로봇의 부족한 자원을 효율적으로 구성하기 위한 다양한 컨텐츠를 지원하게 구성되어 있다. 또한 개발을 용이하게 효과적으로 진행하기 위한 통합 개발 도구가 모든 단계에서 지원하고 있으며, 컴포넌트의 효과적인 활용과 소프트웨어 시장 활성화를 위한 평가 시스템을 구성하고 있다. 따라서 전 개발 과정의 프로세스를 지원하고 있으며 단독 로봇과 네트워크 로봇을 통합적으로 지원하고 있다.

주요한 개발 목표는 다음과 같다.

- 컴포넌트 표준 및 참조 컴포넌트 개발
- 로봇 컨텐츠 언어 및 시뮬레이션을 포함한 통합 개발환경 개발
- 신뢰성 높고 안전하고 유연한 로봇 프레임워크 개발
- 신뢰성 높은 서비스 지향 서버 개발
- 로봇 SW 컴포넌트, HW 모듈 및 컨텐츠 시험 및 평가 시스템 개발

그림과 같이 OPRoS는 매우 광범위한 개발 목표를 가지고 있어서 모든 기술을 논하기는 지면이 부족하다. RUPI 서버로 표현된 로봇 서버 기술, 로봇내 메인 보드에 실장되어 동작하는 로봇 소프트웨어 프레임워크 기술과 로봇 컴포넌트 및 태스크 플랫폼이 주요한 기술이 된다. 또한 이와 같은 로봇 소프트웨어 컴포넌트를 체계적으로 개발하기 위한 개발 방법과 평가 방법론 그리고 개발 환경이 주요 기술이 된다.

먼저 소프트웨어 프레임워크는 모듈 구조의 시스템 구성을 위한 기술로서 모듈 상호간에 하드웨어 및 소프트웨어



#### 4. 결론

지금까지, 로봇 소프트웨어 플랫폼을 생태계 형성을 위한 도구의 용이성이라는 관점에서 살펴보았다. 로봇 산업을 활성화하여야 한다는 측면에서 MS 측의 전략을 면밀히 검토해 보아야 하는데 현재 사업적으로 로봇 소프트웨어 도구가 로봇 산업을 활성화하기에 미약하다고 판단하고 있다. 이것은 국가적으로 진행하고 있는 로봇 소프트웨어 플랫폼 과제에서 신중히 검토해 보아야 한다. 무엇보다도 현재 국내 로봇 산업의 개발자와 사용자가 로봇 소프트웨어 플랫폼을 필요로 하고 있는가 하는 점이며, 기술적으로도 적용이 가능한가 하는 점이다. 또한 모든 개발의 목표가 산업화를 활성화하기 위한 표준화되고 사용이 편리하여 현재 로봇 산업체의 가장 큰 문제점이 소프트웨어 개발과 통합 작업을 손쉽게 지원하여 궁극적으로는 로봇 산업을 활성화하여 로봇 소프트웨어 컴포넌트 자체가 또 하나의 산업으로 성장하는 것이다. 이러한 생태계가 선순환 구조를 이루어갈 때 로봇 산업과 이를 지원하는 로봇 소프트웨어 플랫폼이 동시에 발전해 갈 수 있을 것이다.

결론적으로 로봇 소프트웨어 플랫폼이 로봇 산업의 생태계를 활성화 시키는 쪽으로 기능이 집중되어야 하며, 특히 독립적이고 표준화된 플랫폼을 활용하여 많은 사용자가 다양한 응용 프로그램을 작성하고 이와 같은 유용한 응용을 이용하기 위하여 로봇을 구매하는 생태계가 형성되어야 한다.

#### 참고문헌

[1] Riehle, Dirk, "Framework Design: A Role Modeling Approach", Swiss Federal Institute of Technology, 2000

[2] Michael Somby, "Software platforms for service robotics", <http://www.linuxdevices.com>, 2008

[3] MSRS, Microsoft, <http://www.microsoft.com/robotics>

[4] ERSP, Evolution Robotics, <http://www.evolution.com>

[5] OROCOS, <http://www.oroocos.org>

[6] OpenRTM-aist, <http://www.openrtm.org>

[7] OPRoS, <http://www.opros.or.kr>

[8] 김영준, "Architecture and Inter-Operability of MSRS", Int. Forum on Robot Unified Software Platform, 포항, 2007.11.21

[9] 최병욱, "로봇 소프트웨어 프레임워크 연구동향", 4회 한국지능로봇종합학술대회, 튜토리얼, 특별세션, 2009.7

[10] ORCA Robotics Project, <http://orca-robotics.sourceforge.net>

[11] Alexei Makarenko, "Orca Project and Thin Robotic Software Frameworks", Int. Forum on Robot Unified Software Platform, 포항, 2007.11.21

[12] Tetsuo KOTOKU, "Robotics Technology Component and RT-Middleware", Int' Forum on Robot Unified Software Platform, 포항, 2007.11.21

[13] OMG robotics DTF, <http://robotics.omg.org>

[14] B.Song, S.Jung,C. Jang and S.Kim, "An Introduction to Robot Component Model for OPRoS (Open Platform for Robotic Services), Intl. Conf. on SMPAR2008, pp.592-603, 2008

[15] 로보메이션 홈페이지, <http://www.robomation.kr>

[16] Il Hong Suh, Kyung Jin Kim, Sung Hoon Kim, Sang Rok Oh, "A novel real-time control architecture for Internet-based Thin-client robot; Simulacrum-based approach", ICRA 2008

[17] 최윤석, "플랫폼 기반의 비즈니스 동향 및 전략", SW Insight 2009



#### 최 병 욱

1986 한국항공대학교 항공전자공학과 (공학사)  
 1988 한국과학기술원 전기및전자공학과 (공학석사)  
 1992 한국과학기술원 전기및전자 (공학박사)

1988~2000 LG산전주식회사 엘리베이터 연구실장 및 임베디드 시스템 연구팀장  
 2000~2005 선문대학교 제어계측공학과 부교수  
 2007~2008 Nanyang Technological University, Senior Fellow  
 2005~현재 서울산업대학교 전기공학과 교수  
 관심분야: 임베디드 시스템, 임베디드 리눅스, 실시간 운영체제, 지능형 로봇 응용, 소프트웨어 플랫폼