

## 왜 로봇 소프트웨어 플랫폼을 써야 하는가

**최근 들어 로봇틱스 분야에서** 플랫폼 붐이 일고 있다. 플랫폼은 소프트웨어 플랫폼과 하드웨어 플랫폼으로 나눌 수 있다. 로봇 소프트웨어 플랫폼이란 로봇 응용프로그램을 개발할 때 필요한 하드웨어 추상화, 하위 디바이스 제어, 로봇틱스에서 많이 사용되는 센싱, 인식, 동시적 자기 위치 추정과 지도 작성(SLAM), 내비게이션(Navigation), 매니퓰레이션(Manipulation) 등의 기능 구현은 물론이고, 패키지 관리, 개발 환경에 필요한 라이브러리와 다양한 개발/디버깅 도구 등을 포함하는 것을 말한다. 로봇 하드웨어 플랫폼의 경우 모바일 로봇, 드론, 휴머노이드 형태의 연구용 하드웨어 플랫폼뿐만 아니라 소프트뱅크의 페퍼, MIT 미디어랩의 지보 등 상업성 제품들도 출시를 코앞에 두고 있다.

**주목할 점은** 이 하드웨어들도 앞서 말한 소프트웨어 플랫폼과 연계되어 추상화가 이루어지고 있다는 점이다. 이는 하드웨어에 대한 전문 지식이 없어도 소프트웨어 플랫폼을 이용하여 응용 프로그램을 작성할 수 있다는 것을 의미한다. 이는 최신 스마트폰의 하드웨어 구성이나 세부 내용을 몰라도 애플리케이션(App)을 작성할 수 있는 것과 마찬가지이다. 또한, 로봇 개발자가 하드웨어 설계부터 소프트웨어 설계까지 도맡아 하던 이전 작업 프로세스와 달리, 더 많은 소프트웨어 인력들이 로봇 응용 프로그램 개발에 참여할 수 있다. 즉, 소프트웨어 플랫폼 덕분에 많은 이들이 로봇 개발에 동참할 수 있게 되었고, 소프트웨어 플랫폼에서 제안하는 인터페이스에 맞춰 로봇 하드웨어가 설계되고 있다.

**이러한 소프트웨어 플랫폼에는** 대표적으로 로봇 운영체제라 불리는 ROS(Robot Operating System)와 일본의 오픈 로봇틱스 테크놀로지 미들웨어(OpenRTM), 유럽의 실시간 제어 중심의 OROCOS, 한국의 OPRoS 등이 있다. 각자 이름은 다르지만, 로봇 소프트웨어 플랫폼이 만들어진 근본적인 이유는 로봇 소프트웨어가 너무 다양하고 복잡해서 발생하는 여러 문제를 전 세계의 로봇 연구자가 서로 협업하여 해결하려는 것이다. 이 중 가장 많이 사용되고 있는 로봇 소프트웨어 플랫폼은 이 책에서 중점적으로 다룬 로봇 운영 체제 ROS이다.

**예를 들어,** 로봇이 주변 상황을 인식하는 기능을 구현할 때 하드웨어가 다양하고 실생활에 직접 사용된다는 점은 어려움으로 작용한다. 사람의 입장에서는 매우 사소한 일이라고 하더라도 로봇의 입장에서는 센싱, 인식, 지도 작성, 모션 플래닝 등의 기능을 구현해야 하는데 개별 연구실, 회사에서는 이 모든 부분을 처리하기 힘들다. 하지만 전 세계의 관련 종사자가 각자 자신 있는 부분을 공유하여 다른 그룹에서 이를 사용할 수 있게 한다면 이야기는 달라진다. 예를 들어, 소셜펀드 킥스타터와 CES2015에서 주목받은 로봇베이스(Robotbase)라는 로봇 전문기업은 최근에 로봇베이스 퍼스널 로봇(Robotbase Personal Robot)을 개발하여 소셜펀드에서 론칭에 성공하였다. 로봇베이스사의 경우 자신들이 강한 얼굴인식과 물체인식에 집중하고 모바일 로봇은 유진로봇의 거북이, 액추에이터로는 로보티즈의 다이내믹셀을 이용하였고 장애물 인식,

내비게이션, 모터 드라이브 등은 모두 ROS의 공개 패키지를 사용했다는 점에서 협업 가능성에 대한 실례를 보여주었다.

**그럼 “왜 로봇 소프트웨어 플랫폼을 써야 하는가?”에 대해 알아보자.**

**첫째, 프로그램의 재사용성이다.** 자신이 개발하고자 하는 부분에 집중하고 나머지 기능에 대해서는 관련 패키지를 내려 받아서 사용할 수 있다. 마찬가지로 자신이 개발한 프로그램은 다른 이들이 사용할 수 있도록 공유할 수도 있다. 예를 들어, 미국 NASA의 경우 국제 우주 정거장에서 사용하는 로보노트2(Robonaut2) 로봇 제어를 위하여 자체 프로그램 이외에도 다양한 드라이버 기능과 멀티 플랫폼에서 사용 가능한 ROS와 실시간 제어, 메시지 통신 복구, 신뢰성을 갖춘 OROCOS를 혼용함으로써 우주에서 임무를 수행할 수 있었다고 한다. 앞서 소개한 로봇베이스사의 경우도 재사용성을 충분히 살린 예이다.

**둘째, 통신 기반 프로그램이다.** 흔히, 하나의 서비스를 제공하기 위하여 센서나 액추에이터 단의 드라이브부터 센싱, 인식, 동작까지 하나의 프레임에서 프로그램을 작성하는 것이 많은데 로봇 소프트웨어의 재사용을 위해서는 이를 각각 처리 프로세서의 목적에 따라 작게 나누게 된다. 플랫폼마다 이를 컴포넌트화 혹은 노드 패키지화라고 한다. 최소 실행 단위로 나뉜 프로그램은 나누어진 노드끼리 데이터를 주고받아야 하는데 플랫폼들은 이 데이터 통신에 대한 전반적인 사항을 모두 갖추고 있다. 각 노드는 하드웨어 의존성을 떠나 네트워크에서 통신을 제공함으로써 네트워크 프로그래밍이 가능하게 되고 로보틱스에서 흔히 다루는 원격제어에서 매우 유용하다. 또한, 최근 사물인터넷(IoT) 개념과도 일치하여 사물인터넷 플랫폼 대신 사용할 수 있으며 최소 실행 단위로 나뉜 프로그램은 작은 단위로 디버깅할 수 있어서 오류를 찾아낼 때도 매우 유용하다.

**셋째, 개발 도구 지원이다.** ROS의 경우 디버깅 관련 도구, 2차원 플롯과 3차원 시각화 도구를 제공함으로써 로봇 개발에 필요한 개발 도구를 직접 개발할 필요 없이 바로 사용할 수 있다. 예를 들어, 로봇 개발에 있어서 로봇의 모델을 시각화하는 경우가 많은데 이를 정해진 포맷에 맞추기만 하면 로봇의 모델을 직접 확인할 수 있을 뿐만 아니라 3차원 시뮬레이터도 제공하기 때문에 시뮬레이션으로의 확장도 쉽다. 또한, 요즘 주목 받는 MS사의 키넥트 등으로 얻은 3차원 거리 정보는 점 군을 나타내는 포인트 클라우드 형태로 쉽게 변환하여 보여준다. 그 이외에도 실험에서 사용된 데이터는 녹화할 수 있으므로 언제든지 필요할 때 재생하여 실험할 때의 상황을 그대로 재현할 수도 있다. 이처럼 로봇 개발에 꼭 필요한 소프트웨어 도구를 제공하여 개발 편의성을 극대화한 점이 중요한 원인 중의 하나이다.

**넷째, 생태계 조성이다.** 스마트폰 혁명은 안드로이드와 iOS 등 소프트웨어 플랫폼이 만든 생태계가 있기 때문이라고 한다. 로봇 분야도 마찬가지로 흐름으로 이어 가고 있다. 처음에는 각종 하드웨어 기술들이 넘쳐흘렀으나, 이를 통합해줄 운영체제가 없었다. 이 상황에서 앞서 설명했던

것과 같이 다양한 소프트웨어 플랫폼이 등장했고, 가장 주목 받은 ROS의 경우 이제 그 생태계의 틀을 갖추기 시작했다. 그리고 로봇과 센서 회사처럼 로봇 관련 하드웨어 분야의 개발자, ROS 개발 운용팀, 응용 소프트웨어 개발자, 사용자 모두가 웃을 수 있는 생태계를 만들어 가고 있다. 아직 그 시작은 미미하지만, 점점 늘고 있는 사용자와 로봇 관련 회사들 그리고 급격히 늘고 있는 관련 도구 및 라이브러리를 볼 때 머지않아 생태계가 원만하게 돌아갈 것이라고 기대해본다.

**다섯째, 활성화된 커뮤니티이다.** 이 부분이 가장 중요하지 않을까 싶다. 지금까지 닫혀 있던 로봇 학계, 로봇 업계는 위에서 언급한 기능들로 인하여 서로의 협업을 중시하는 방향으로 나아가고 있고 그 목적이야 서로 다를 수 있겠지만 이러한 소프트웨어 플랫폼을 통하여 협업이 실제로 이루어지고 있다. 이 중심에는 오픈 소스 소프트웨어 플랫폼의 커뮤니티가 있다. 예를 들어 ROS의 경우에는 2015년을 기준으로 자발적으로 5,000개 이상의 패키지들이 개발되어 공유되고 있고, 그 사용 방법을 설명한 위키 페이지가 사용자들의 개별적인 참여로 16,000페이지를 넘어서고 있다. 그리고 커뮤니티에서 매우 중요한 질의응답의 경우 24,000건 이상이 오고 가며 상생의 커뮤니티를 만들어 가고 있다. 이는 단순히 사용법에 대한 토론을 넘어서 로보틱스 소프트웨어에서 필요한 구성요소를 찾아내고 규칙을 만들고 있다. 나아가 로보틱스의 발전을 위해 로봇의 소프트웨어가 갖추어야 할 부분에 대해서 고민하고 부족한 부분은 서로 협업을 통하여 다수가 하나의 퍼즐을 맞추는 식으로 발전하고 있다.

**로봇공학은 미래 산업,** 차세대 성장 동력이라는 이름으로 산업용 로봇을 제외한 어떠한 비즈니스 모델도 없음에도 불구하고 장밋빛 미래를 바라며 기대만을 한껏 받고 있다. 문제는 이러한 이야기는 10년 전이나 지금이나 마찬가지이지만 이렇다 할 답을 못 얻고 있다는 것이다. 그 이유는 무엇일까? 다양한 의견들이 있겠지만 로봇공학은 아직 비즈니스 모델로 만들기에는 부족한 부분이 많다는 것이고 아직 풀어야 할 숙제가 많다고 생각한다. 이 해결책으로 국가를 뛰어넘는 초월적인 협업이 이루어져야 한다고 생각하고 이는 소프트웨어 플랫폼과 이를 지탱하는 커뮤니티가 해결할 수 있다고 생각한다. ROS의 경우 학계 연구자, 산업현장의 개발자, 그리고 취미로 활동하는 하비스트(hobbyist)까지 개발에 참여하고 있다. 더욱이 로봇 전공자뿐만 아니라 네트워크 전문가, 컴퓨터 사이언스, 컴퓨터 비전 분야의 사람들도 대거 참여하고 있어서 로봇 분야뿐만 아니라 다양한 배경의 지식이 모여 융합을 이루고 있어서 로봇공학이 지금까지와는 다른 양상으로 발전할 것으로 예상하며, 개방과 협업을 통하여 지금까지 풀지 못한 숙제들을 풀 수 있을 것으로 생각한다. 또한, 이는 로봇 개발이 급속도로 발전할 수 있는 계기가 될 수 있을 것이다.

로보티즈 선임연구원

pyo@robotis.com

표윤석