레고 NXT의 제어를 위한 로보이드 스튜디오의 응용

임준원, 이상훈, 김종복, 김경진, 서일홍
임준원, 김종복, 서일홍 : 한양대학교 전자컴퓨터통신공학과
이상훈 : 한양대학교 BK21 수요지향적 정보기술 전문인력양성 사업단
김경진 : (주) 로보메이션

Application of ROBOID Studio to the control of LEGO NXT

J. Lim, S. Lee, J. Kim, K. Kim, I. H. Suh
J. Lim, J. Kim, I. H. Suh : Dept. of Electronics and Computer Engineering, Hanyang University
S. Lee : BK21 Advanced IT Education Program on Industrial Demand, Hanyang University
K. Kim : Robomation, Co., Ltd.

Abstract - 네트워크 기반의 씬 클라이언트 로봇은 내장된 리소스가 부족하지만 네트워크를 통해 이를 극복하려는 로봇이다. 씬 클라이언트 로봇에는 부족한 내장 리소스를 네트워크를 통해 효과적으로 운영하기 위한 디바이스 맵 프로토콜과 로봇 제어 및 개발 플랫폼으로써 로보이드 스튜디오가 제안되어 있다. 현재 로보이드 스튜디오는 공개 소프트웨어 방식으로 다양한 로봇을 지원하지만, 아직은 실제 지원되는 로봇의 종류가 부족하다. 따라서 새로운 로봇에 대한 이식 지원 방법과 다양한 이식 환경 속에서의 문제점을 찾아 해결해야 한다. 우리는 첫 번째로 씬 클라이언트 로봇보다는 내장 리소스가 풍부하며, 광범위한 사용자층을 갖고 있는 LEGO NXT를 선택하여 로보이드 스튜디오 운영 및 개발 환경의 이식 절차를 제안하고, 로보이드 스튜디오 프레임워크가 제공하는 네트워크 기반 로봇 제어 기술의 성능을 검증한다.

1. 서 론

개인용 지능형 서비스 로봇은 제조와 공장 자동화 등의 정해진 일을 반복 수행하는 산업용 로봇과는 달리 인간의 생활에 밀착된 서비스인 오락, 교육, 가사 등의 목적으로 소비자들에게 구매되는 로봇이다. 현재는 토이 로봇과 청소 로봇 등으로 제한적이지만 다양한 서비스와 기능, 그리고 경쟁력 있는 제품 가격이 뒷받침해준다면 앞으로 여러 분야에서 급속도로 발전할 것으로 예측된다.

최근 빠르게 발전하고 있는 IT(Information Technology) 기술과 인프라를 바탕으로 다양한 서비스를 제공하는 네트워크 로봇에 대한 관심이 증가되고 있다. 사용자에게 반복적으로 같은 서비스 콘텐츠를 보여주는 것은 흥미를 감소시키므로, 새로운 서비스의 대한 요구를 만들어 내야한다 [2]. 기존 독립형 로봇은 다양한 서비스의 대한 요구를 충족시키지 못한 것에 비해, 네트워크 로봇은 인터넷을 통해 새로운 서비스를 끊임없이 제공 할 수 있기 때문에 이런 결점들을 극복할 수 있다. 특히 네트워크 로봇 종류 중 하나인 씬 클라이언트 로봇은 비용, 접근성, 편의성 등에 있어서 상당히 경쟁력이 있으며, 많은 계산량이 필요한 얼굴인식, 음성인식 등의 지능적 서비스 등도 원격 서버에게 전달하여 효율적으로 처리할 수 있다. 이런 이점에도 불구하고 네트워크 로봇을 위한 어플리케이션을 개발하는데 다음과 같은 몇 가지 어려움이 있다[2].

- 웹 서비스 사용자들은 멀티미디어 콘텐츠에 익숙하다. 그래서 사용자들은 멀티미디어 데이터를 로봇의 모션 데이터처럼 로봇 콘텐츠에 포함 되는 것을 원한다.
- 로봇에 각 부분은 다른 부분들과 동기화가 필요하다. 이 점을 위해서는 여러 센서와 디바이스의 정보를 동시에 전달할 수 있는 프로토콜이 필요하다.
- 독립형 로봇에 비해 전송 지연, 대역폭의 제한, 패킷 손실[3], [4], [5], [6] 등과 같은 제어의 어려움이 있다.
- 일반적인 소프트웨어 플랫폼은 로봇 콘텐츠를 공유하거나 반복적인 일을 효율적으로 관리하는 점이 부족하여 개발 비용감소가 어렵다.

이러한 문제들을 위하여 우리는 씬 클라이언트 로봇을 위한 프레임워크로 로보이드 스튜디오와 디바이스 맵 프로토콜을 소개한다. 제안된 구조는 실시간으로 원격제어를 통하여 로봇의 상태를 완벽하게 전달하고, 디바이스 맵 프로토콜은 신뢰성 있고 강인한 제어를 할 수 있도록 다양하고 유연성 있는 패킷 구조와 크기를 갖고 있다.

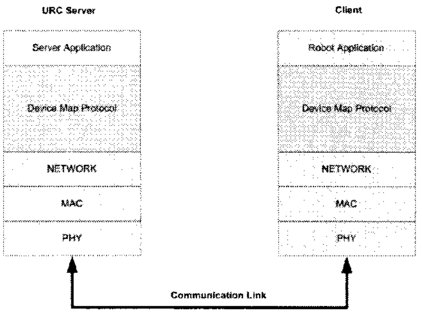
우리는 이러한 프레임워크에 모든 씬 클라이언트 로봇을 모델링 과정을 통하여 자동적으로 이식시키기 위하여 프레임워크와 로봇 사이의 미들웨어를 개발하였다. 씬 클라이언트 로봇의 하드웨어 추상화를 위한 디바이스 컴포넌트와 디바이스 맵 프로토콜 기반의 URC 서버 클라이언트 통신 프로토콜을 이용하여 시범적으로 LEGO MINDSTORM NXT 로봇을 적용시켰고, 로보이드 스튜디오의 호환성 확장을 위한 미들웨어의 필요성과 그 기능들을 살펴 볼 것이다.

본 논문의 2.1장에서는 로보이드 스튜디오와 디바이스 맵 프로토콜의 구조와 구성요소들에 대하여 설명할 것이다. 2.2장에서는 로봇을 이 프레임워크로 이식 할 경우, 이식 환경에서의 문제점을 살펴보고 해결책으로써 우리가 제안한 미들웨어와 그 이식 절차를 보일 것이다. 2.3장에서는 개발한 미들웨어를 적용한 실험 및 결과를 보일 것이다. 3장에서는 결론과 앞으로 해결해야 할 과제에 대해서 언급할 것이다.

2. 본 론

2.1 로보이드 스튜디오

로보이드 스튜디오는 씬 클라이언트 로봇 소프트웨어의 개발을 위하여 플랫폼 독립성, 비쥬얼 프로그래밍 환경, 동시성, 분산환경, 컴포넌트, 로봇 콘텐츠, 스크립트 언어지원, 시뮬레이크럼[7] 등의 기능을 제공하는 프레임워크이다. 아래 그림1에 나와 있는 것과 같이, 디바이스 맵 프로토콜을 이용하여 통신을 한다.



<그림 1> 통신 규약의 범위

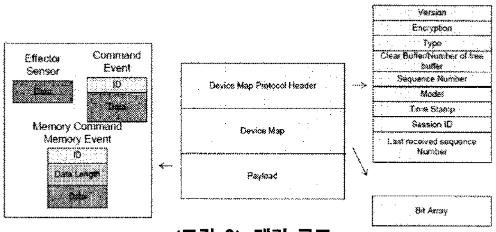
2.1.1 네트워크 기반의 씬 클라이언트 로봇(Thin-Client Robot)

씬 클라이언트 로봇은 하드웨어 자원이 부족한 로봇으로, 네트워크 연결을 통해 서버의 도움으로 기능을 수행할 수 있는 로봇을 말한다. 네트워크에 연결되지 않는 경우, 씬 클라이언트 로봇이 수행할 수 있는 기능은 거의 없다.[7] 하지만 센서 및 디바이스의 정보가 디바이스 맵 프로토콜로 생성된 패킷을 통하여 원격 서버에 전송된다면 물체 인식, 얼굴 인식, 사람 감지 등의 상당한 계산 능력이 필요한 작업들도 처리할 수 있기 때문에 효율적인 제어가 가능하다. 현재 씬 클라이언트 로봇으로는 (주)로보메이션에서 출시한 펠리카노이드 등이 있으며, 본 논문에서는 씬 클라이언트 로봇을 위한 플랫폼인 로보이드 스튜디오가 모든 네트워크 로봇을 제어하도록 이식성을 증대하기 위하여 LEGO NXT 로봇과의 상호호환성을 보인다.

2.1.2 디바이스 맵

디바이스 맵 프로토콜을 따르는 패킷포맷의 구조는 그림 2와 같다. 패킷포맷은 모두 3 부분으로 나뉘어지며 헤더 부분과 디바이스 맵 부분은 데이터 전송 시 항상 포함되어야 한다. 마지막 부분을 구성하는 페이로드는 디바이스 맵의 내용에 따라서 크기가 가변적이다[7].

디바이스 맵 부분에는 이펙터, 커맨드, 센서, 이벤트[7] 등의 디바이스와 센서의 대한 정보의 활성화 상태를 1과 0으로 조합된 비트 배열로 나타낸다. 페이로드에는 이런 디바이스 맵의 활성화 상태를 기준으로 각 디바이스의 값을 할당한다. 이렇게 구성된 패킷은 네트워크 계층과 MAC, PHY 층을 거쳐서 로봇 또는 서버로 전달된다.



<그림 2> 패킷 구조

2.2 이식 환경

2.2.1 이식 환경속의 문제점

로보이드 스튜디오는 새로운 로봇의 이식을 위하여, 로봇의 하드웨어 추상화를 위한 디바이스 컴포넌트와 각 디바이스의 모델링 과정을 위한 모델링 툴을 제공한다. 이러한 점들과 언급했던 플랫폼 독립성, 분산 환경, 동시성 등 로보이드 스튜디오의 다양한 이점에도 불구하고 새로운 로봇을 이 프레임워크에 이식시켜 제어하기 위해서는 디바이스 맵 프로토콜에 따라 패킷의 해석 및 변환, 로봇과 서버사이의 통신설정, 해석된 패킷 정보에 따른 로봇 제어 등의 기능을 담당하여 처리 할 중간 매개체가 필요하다. 즉, 사용자가 직접 해당 로봇 제어 서비스 모듈을 개발해야 되는데, 이것은 로보틱스 어플리케이션 개발자 뿐 만이 아니라 일반인에게도 로보이드 스튜디오의 편리한 사용을 제공하고자 했던 원래 의도에 맞지 않게 된다.

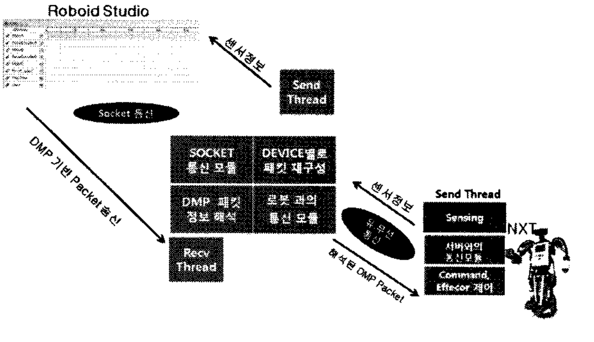
2.2.2 이식 지원 방법 및 절차 제안

우리는 네트워크 로봇의 자동적인 이식 지원 방법으로 그림 3과 같은 구성의 미들웨어를 개발하였다. 실제 로봇은 로보이드 스튜디오에서 모델링되어 로봇 모델 파일을 생성하고 가상 로봇으로 만든다. 비쥬얼 프로그래밍 환경을 통하여 생성된 콘텐츠들은 기본적으로 특정 포트로 소켓 통신을 통하여 해당 로봇의 모델 파일을 참고로 하여 디바이스 맵 프로토콜의 규격을 따르는 패킷을 송신하다. 미들웨어의 개발 목적은 실시간으로 이 패킷들을 해석 및 재구성하여 로봇에게 다시 패킷을 전송하고, 반대로 로봇의 센서 디바이스들의 정보 역시 디바이스 맵 프로토콜 규격에 따라 패킷으로 생성하여 로보이드 스튜디오에서 센싱 정보를 읽어 들여 제어할 수 있도록 하기 위함이다. 또한 로봇과 서버와의 통신 연결 기능들을 제공하며, JAVA 언어로 개발되었으며 컴포넌트 모델링을 지향하는 방식으로 개발하였다. 레고 NXT 실험에서는 로봇과 서버의 통신을 블루투스를 사용하였으며, 미들웨어의 기능을 정리해 보면 다음과 같다.

- 모델링 된 로봇 모델 파일에 따라서 패킷 해석 및 생성한다.
- 로보이드 스튜디오에서는 20ms 간격으로 패킷을 송신하는데, 동기화를 위해서 2개의 recv, send 쓰레드를 동작시켜서 실시간 환경에서 로봇의 강인한 행동제어를 지원한다.
- 로봇 모델 파일은 XML로 구성되어 있다. 모델에 따른 가변적인 제어를 위하여 로봇 어플리케이션에 알맞은 이벤트 중심의 인터페이스를 갖는 SAX를 이용하여 모델 파일을 파싱한다.
- 패킷 모니터링 툴을 제공한다.

그림 3을 보면 미들웨어는 각 모듈별로 나뉘어져 있다. 다른 어느 로봇이라도 모델 파일에 따라 패킷 재구성 모듈을 변경하고, 로봇과 서버와의 통신 모듈 역시 로봇에 따른 다양한 통신 방법으로 변경한다면 나머지 모듈들의 재사용으로 인해 최소한의 변경으로 로보이드 플랫폼에 모든 네트워크 로봇을 적용 시킬 수 있다. 다음과 같은 이식절차를 제안하였다.

ⓐ 로봇을 가상 로봇으로 모델링 한다.
ⓑ 해당 로봇의 통신 방법으로 미들웨어의 통신 모듈을 변경한다.
ⓒ 해당 로봇의 모델링 정보에 따라 제어 가능하도록 패킷 재구성 모듈 변경한다.
ⓓ 로봇의 제어로직, API를 호출할 수 있도록 Sensing과 커맨드, 이펙트 제어 모듈을 변경한다.



<그림 3> 미들웨어 구성

2.3 실험 및 결과

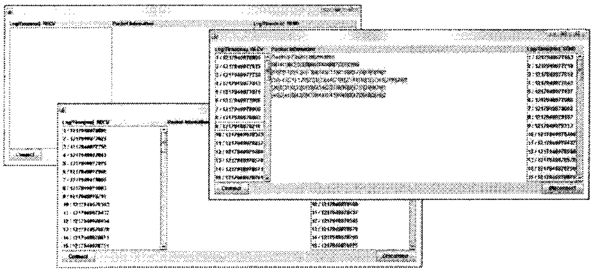
2.3.1 LEGO MINDSTORM NXT

레고 마인드스톰 NXT는 레고사에서 MIT와 산학협동으로 만들어낸 제품으로 프로그래밍 가능한 브릭을 지향하고 있다[1]. 사용자의 컴퓨터에서 명령을 내리면 유선(USB) 또는 무선(Bluetooth)으로 명령들이 전송되어 NXT 브릭의 메모리에 저장된다. NXT의 입력 단에는 소리, 빛 등을 감지하는 각종 센서가 연결되고 출력 단에는 센서와 메모리에 저장된 명령을 종합하여 실제 구동되는 모터 등의 디바이스가 연결된다.

NXT는 두뇌라 할 수 있는 브릭과 각종 센서, 그리고 3개의 모터로 구성되어 있다. 현재 NXT는 레고사에서 제공하는 LSOFT, 내쇼날 인스트루먼트사의 LabVIEW, 마이크로 소프트사의 MSRDS 등의 플랫폼에서 제어가 가능하며, 우리는 현재 국내 플랫폼인 로보이드 스튜디오에서 제어 가능하도록 확장, 이식 시키는 시도를 하였다.

2.3.2 패킷 모니터링 툴

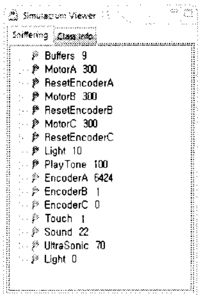
로보틱스 어플리케이션 개발자들을 위하여 미들웨어와 함께 그림 4에서 보이는 패킷 모니터링 툴을 제공하여 패킷의 내용과 시간정보를 손쉽게 볼 수 있게 하였다. 로봇과 서버간의 송수신되는 디바이스 맵 프로토콜 규격에 따라 생성 된 패킷들의 해석되지 않은 본래의 정보를 실시간으로 확인 할 수 있어 개발 및 디버깅을 수월하게 수행하도록 도움을 준다.



<그림 4> 패킷 모니터링 툴

2.3.3 실험 결과

그림 5을 보면, 현재 로봇의 각종 센서 및 디바이스의 값을 확인 할 수 있도록 로보이드 스튜디오에서 제공하는 시뮬레이크럼에 실시간으로 NXT의 값들이 변경되고 있는 모습이다. NXT의 제어 및 동작은 물론, 패킷 전달 역시 전송 지연 없이 로보이드 스튜디오에 적용되어 실시간 환경에서 신뢰성 있게 동작하였다는 것을 알 수 있다. 서버 측에서 항상 20ms 단위로 패킷을 송신하는 점과 NXT 자체가 20ms 단위로 명령을 받는 것이 허용되지 않는 것은 미들웨어 측에서 패킷을 조절하여 동기화가 맞도록 제어하였고, 중요한 명령 또는 데이터는 수신확인을 하여 재전송 기능을 제공하는 커맨드[7]를 사용하였기 때문에 패킷 손실문제도 존재하지 않았다.



<그림 5> NXT의 센서 및 디바이스 값

3. 결 론

실험을 통해, 우리가 제안한 이식 절차와 미들웨어를 통하여 로보이드 스튜디오와 모든 네트워크 로봇 사이의 호환성 있는 제어가 가능한 것을 보였고, 이것은 로봇 개발자와 일반인 모두에게 편리한 로봇 프로그래밍 및 다양한 콘텐츠 개발을 가능하게 할 것이다. 향후에는 현재 XML로 구성되어 있는 로봇 모델 파일을 UML 로봇 프로파일로 대체 할 것이고, 미들웨어의 구성을 컴포넌트 모델링 관점에서 완벽히 접근하여 모듈 추가, 수정과 컴포넌트의 재사용등을 가능하게 하여 네트워크 로봇의 새로운 패러다임을 도입하는 로봇 소프트웨어 체계를 완성하고, 로봇의 대중화 및 상업화에 활용 되어 질 것으로 기대된다.

[참 고 문 헌]

[1] Brian Bagnall, Maximum LEGO NXT, VARIANT PRESS, 2007
[2] K. J. Kim, I. H. Suh, K. H. Park, "Roboid Studio: A Design Framework for Thin-Client Network Robots," in Proceedings of the 2008 IEEE International Conference on Advanced Robotics and its Social Impacts, August 23-25, 2008.
[3] K. J. Kim, I. H. Suh, S. H. Kim, and S. R. Oh, "A novel real-time control architecture for internet-based thin-client robot: Simulacrum-based approach," in Proceedings of the 2008 IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, May 19-23, 2008.
[4] P. Li, W. Lu, and Z. Sun, "Transport layer protocol reconfiguration for network-based robot control system," in Proceedings of the 2005 IEEE Networking, Sensing and Control(ICNSC2005), March 19-22, 2005, pp. 1049-1053
[5] K. Goldberg, M. Mascha, S. Gentner, N. Rothenberg, C. Sutter, and J. Wiegley, "Desktop teleoperation via the World Wide Web," in Proceedings of the 1995 IEEE International Conference on Robotics and Automation, vol. 1, May 21-27, 1995, pp. 654-659.
[6] H. Hu, L. Yu, P. W. Tsui, and Q. Zhou, "Internet-based robotic systems for teleoperation," International Journal of Assembly Automation, vol. 21, no. 2, pp. 143-151, 2001.
[7] RUPI 사무국, "씬 클라이언트 로봇을 위한 디바이스 맵 기반의 URC 서버/클라이언트 통신 프로토콜", 2007.12