한국컴퓨터정보학회 동계학술대회 논문집 제19권 제1호 (2011. 1)

자바를 이용한 오픈 플랫폼 미들웨어의 설계에 관한 연구

박민기*, 장성환°, 정경원*
°*서울과학기술대학교 전자정보공학과
e-mail : mkpark@snut.ac.kr, kzjsh21@snut.ac.kr, for-you-sa@hanmail.net

The study on the design of open platform middleware using JAVA

Min-kee Park*, Sung-hwan Jang°, Kyung-won Jung*
°*Dept. of Electronic and Information Engineering, Seoul National University of Science & Technology

● 요 약 ●

본 논문에서는 다양한 하드웨어를 능동적으로 관리할 수 있는 미들웨어 시스템을 제안한다. 최근 WII Remote, KINECT와 같은 게임기에서 사용되는 하드웨어를 일반 데스크톱에서 사용하고자 하는 노력들이 많이 이루어지고 있다. 뿐만 아니라 RFID, 블루투스, 이동전화 등 기존에 나와 있는 다양한 기기들도 데스크톱에서 이용할 수 있도록 많은 기술적 접근이 이루어지고 있다. 각각의 디바이스를 관리하기 위해서는 그에 맞는 디바이스 드라이버와 API를 만들어야만 한다는 단점이 있다. 제안하는 시스템은 시대적 흐름에 맞추어 다양한 하드웨어 시스템을 효율적으로 관리할 수 있으며, C, C++, C#, JAVA의 다양한 언어로 구현된 어플리케이션을 하나의 미들웨어 위에서 동작시킬 수 있는 큰 장점이 있다. 기존의 미들웨어는 이종 다바이스간의 데이터 호환 및 처리를 수행하였지만 본 논문에서 제안하는 시스템은 이종 플랫폼간의 데이터 호환까지 지원한다.

키워드: 미들웨어(Middleware), 오픈 플랫폼(Open Platform), 자바(JAVA)

Ⅰ. 서 론

최근 WII Remote Control, KINECT Camera, 조이스틱, 블루투스, RFID등 다양한 하드웨어들이 일반 데스크탑에 연결되어 보다 interactive한 환경을 제공하고 있다. 키보드와 마우스로만 컴퓨터의 입력을 수행하던 점에 비해 보다 다양한 기기를 이용할 수 있으므로, 사용자의 보다 다양한 입력을 받아들여 각각의 게임기 혹은 임베디드 시스템에서 이루어지던 일련의 작업들을 데스크탑에서 수행할 수 있게 된다. 이러한 시대적 흐름에 맞추어 본 논문에서는 다양한 하드웨어를 통합 운용할 수 있는 강력한 미들웨어(middleware) 시스템을 제안한다.

본 논문에서 제안하는 미들웨어 시스템은 다양한 하드웨어 기기를 능동적으로 붙일 수 있는 장점이 있다. 다양한 하드웨어를 미들웨어에서 관리하기 때문에 일반 개발자 혹은 사용자는 하드웨어에서 일어나는 일련의 작용들을 모르고도 개발 혹은 일을 수행할 수 있게 된다. 하드웨어를 관리해주는 Layer를 Hardware Application Layer라 칭하며, OS위에서 동작하는 다양한 어플리케이션을 관리하는 Layer를 u-VM API Call Layer라 칭한다. 두 Layer 사이에서 상호적으로 작동하는 부분을 u-VM Engine이라 칭하며, 이 부분이 본 논문에서 말하는 미들웨어이다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로 센서 네트워크간 정보교환을 위한 미들웨어 및 이동기기를 위한 미들웨어에 대해 알아보고, 3장에서는 본 논문에서 제안하는 시스템에 대해 기술하고 제안한 방법을 검증한다. 4장에서는 본 연구의 기대효과 및 향후 연구방향에 대해 기술하고 결론을 맺는다.

Ⅱ. 관련 연구

1. 관련연구

1.1 이종 센서 네트워크간 정보교환

미들웨어에 관한 연구는 국내에서는 활발하게 진행되고 있지 않은 실정이다. 하지만 Sensor Network 혹은 Mobile Device등 최근 상용화 되고 있는 device를 대상으로 다양한 미들웨어 시스템이 제안되고 있다. [1]

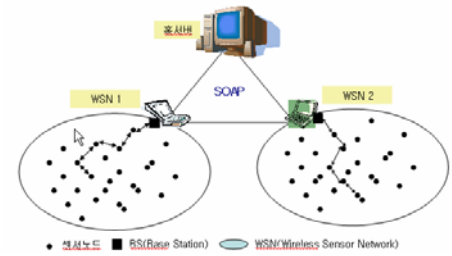

그림 1. 이종 센서 네트워크간 정보교환 구성도
Fig. 1. Structure for information exchange

그림1은 이종 센서 네트워크(sensor network)간 정보교환을 위한 미들웨어 시스템의 구성도이다. WSN1, WSN2에서 받는 데이터를 효과적으로 처리하기 위하여 서버(server)를 구성하고, 그 안에 미들웨어를 내장하여 이종 센서에서 들어오는 데이터도 한곳에서 처리할 수 있도록 하는 것이 목적이다. 다른 센서에서 보내지는 데이터도 함께 처리할 수 있다는 장점이 있지만 센서 네트워크에 한정되어 만들어진 제약이 있다.

1.2 모바일 기기를 이용한 미들웨어

다음으로 살펴볼 부분은 모바일 기기를 위한 미들웨어 시스템이다. JNI Service를 이용하여 컴퓨터와 모바일 기기간의 능동적인 통신이 이루어지도록 제안하고 있지만 이 시스템 또한 모바일 기기에 한정되어 제안하고 있는 시스템이다. [2]

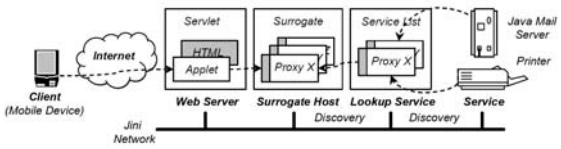

그림 2. 모바일 기기를 위한 미들웨어
Fig. 2. Middleware architecture for mobile device

국내 연구 사례를 살펴보면 한정된 도메인 안에서의 이종 기기간의 통신 혹은 프로세스(process)를 위해 각각의 상황에 맞는 미들웨어 시스템이 제안되고 있다는 것을 알 수 있다.

III. 본 론

본 논문에서 제안하는 시스템은 현재 제안되고 있는 시스템과는 차별화 되어 있다. 먼저, 이종 센서뿐만 아니라 이종 기기 간에서의 통신 및 처리가 가능하다. 또한 JNI를 통한 C언어와 JAVA의 호환이 아니라 C, C++, JAVA, C#, WPF등 다양한 언어를 사용하여 미들웨어에 접근할 수 있도록 한다.

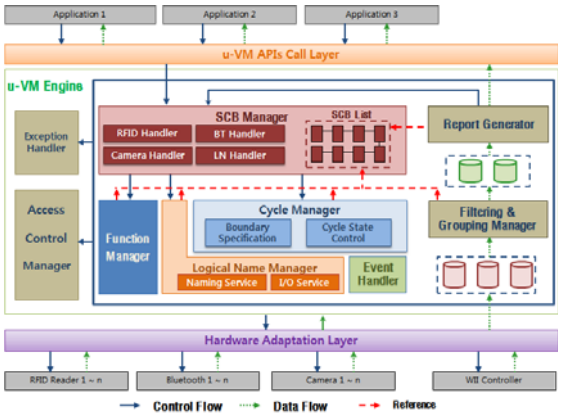

그림 3. 제안하는 시스템 구조
Fig. 3. The proposed system architecture

그림3은 본 논문에서 제안하는 시스템 구조이다. 미들웨어 시스템은 다양한 모듈로 구성되는데, 각각의 모듈이 담당하는 기능에 대해 설명하도록 한다.

먼저 Logical Name Manager부분이다. 다양한 하드웨어에게 하나의 논리적인 이름으로 매치하여 한 번에 기기들을 연결 또는 데이터를 읽거나 쓸 수 있도록 구성한다. Naming Service와 IO Service로 구성된다.

Cycle Manager는 각각의 하드웨어들을 연결 및 해제하는 시점을 지정하는 역할을 담당한다. Boundary Specification과 Cycle State Control로 구성된다.

SCB Manager는 다른 Manager들을 한 번에 관리할 수 있는 블록이며, 각각의 하드웨어를 작동하는 역할을 담당하는 Handler와 LN Handler로 구성되어 있다. 즉, 하드웨어의 개수 N +1개로 구성되어 있다. LN Handler는 각 하드웨어를 1:1또는 1:N으로 매치 시켜주는 역할을 담당한다.

Filtering & Grouping Manager는 시스템 구성도의 오른쪽에 위치한다. 각 하드웨어에서 올라온 정보를 미들웨어에서 쓸모 있는 정보로 가공하는 작업을 수행한다. 예를 들어 카메라의 경우 입력 영상을 버터에 담고 그것을 필터링 해서 손가락을 인식하거나 주사위 등을 인식한 데이터를 다시 버퍼에 저장하는 역할을 하는 것이다.

마지막으로 Report Generator는 Filter Manager에서 저장한 데이터를 가지고 해당 어플리케이션으로 데이터를 전송하는 역할을 수행한다. 이 때 필요한 데이터가 모두 들어갈 수 있도록 보고서 형태로 제작된다.

그림 4. 시스템 Main UI
Fig. 4. System Main UI

이러한 특징을 갖는 미들웨어에 u-VM API Call Layer가 추가적으로 붙게 된다. 상위 어플리케이션에서 사용되는 언어에 상관없이 미들웨어의 특정한 기능을 수행할 수 있도록 해주는 Layer이며, JAVA의 JNI를 이용하여 구현된다. 상위 어플리케이션에서 호환 가능한 언어는 C, C++, JAVA, C#, WPF등이 있다.

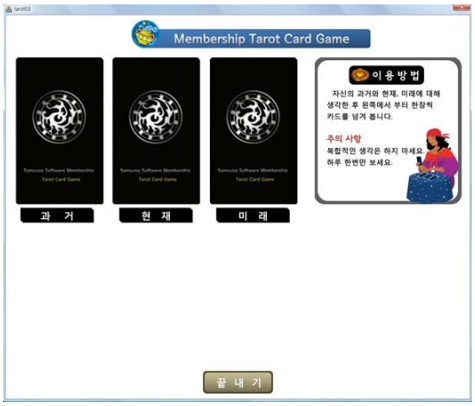

그림 5. MFC용 검증 프로그램 (타로 카드)
Fig. 5. MFC verification program(Tarot Card)

시스템 검증을 위하여 하드웨어에는 WII Remote, Web CAM, RFID Control, Bluetooth, Windows Mobile 6.5 환경의 스마트 폰을 부착하였다. 상위 어플리케이션에서는 데이터가 잘 동작하는지 확인하기 위하여 각각의 언어별로 Tarot Card, Chess, Bluemable게임을 제작하여 미들웨어가 잘 동작하는지 검증하였다.

그림 6. JAVA용 검증 프로그램
Fig. 6. JAVA verification program

그림 7. C#용 검증 프로그램
Fig. 7. C# verification program

미들웨어 성능 검증을 위해서 각 모듈의 초당 처리속도 및 이종 모듈의 초당 처리속도를 측정하였으며, 어플리케이션에서 모듈의 처리량을 측정하였다.

표 1. 하드웨어의 항목별 처리속도 검증
Table 1. Hardware process time verification

항목		처리 속도
RFID		50 tag per sec
Bluetooth		33 packet per sec
Camera		30 frame per sec
Cell phone		20 request per sec
RFID & Bluetooth		48 tag per sec
		27 packet per sec
Bluetooth & Camera		26 packet per sec
		30 frame per sec
All	RFID	48 tag per sec
	Bluetooth	27 packet per sec
	Camera	29 frame per sec
	Cell phone	20 request per sec

표 1에 나타나 있듯이 다수의 하드웨어를 장착하고도 처리 성능의 저하 없이 처리량을 유지할 수 있음을 알 수 있다. 검증은

Intel Quad Core 2.5GHz, 3GB RAM PC환경을 사용하였다.

그림 8은 각 모듈간의 데이터 호환을 검증하는 화면을 나타낸 것이다.

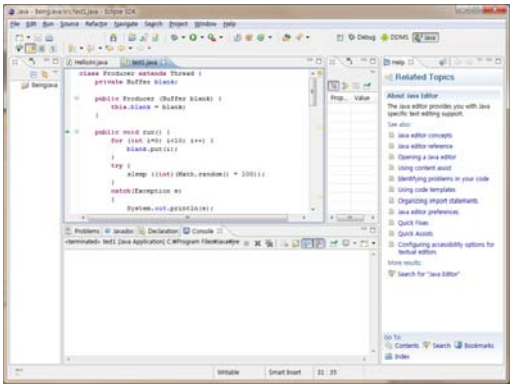

그림 8. 미들웨어 검증 화면
Fig. 8. Middleware verification display

Ⅳ. 결 론

본 논문에서 제안하는 시스템을 이용하여 다양한 하드웨어 기기를 손쉽게 제어할 수 있는 Middleware를 제안하고 구현하였다. 표 1에서 확인하는 바와 같이 다수의 하드웨어를 손쉽게 붙일 수 있으며, 하드웨어의 개수가 늘어나더라도 속도의 저하가 심하지 않은 것으로 나타났다. 본 연구에서 제안한 방법은 다음과 같은 특징을 가진다.

Client는 H/W 또는 기반 소프트웨어를 몰라도 API만을 이용하여 하드웨어나 영상처리 솔루션에 접근할 수 있다. 또한 다양한 Client의 요청을 동시에 처리할 수 있으며, 각각의 Event는 처리 시작과 종료시간을 Handler를 통해 제어할 수 있다. Application은 callback 함수만 만들어 하드웨어에서 올라오는 데이터를 사용할 수 있으며, 물리적으로 떨어진 지역에 있는 하드웨어에도 접근이 가능하다.

앞으로 출현되는 안드로이드, Windows Mobile 7 등 다양한 모바일 기기들과 나탈 프로젝트에 사용되는 KINECT 카메라와 같은 하드웨어에 능동적인 이용이 기대된다.

참고문헌

[1] Manuel Román, Christopher Hess, “A Middleware Infrastructure to Enable Active Spaces”, IEEE Pervasive pp. 154~158, Nov. 2002.

[2] T Gu, HK Pung, “A service-oriented middleware for building context-aware services” Journal of Network and Computer, pp.76~79, 2005.

[3] Sang-tae Kim, “Middleware model design analysis for Mobile Device” Korea Information and Science academy pp.137~142, 2003

[4] Sang-im An, “Middleware Architecture for Information Exchange in Heterogeneous Wireless Sensor Networks”, Korea Information and Science academy pp.352~359, 2005.