

OSGi 기반의 이동 에이전트 서버의 설계 및 구현

박종열⁰ 백의현
한국전자통신연구원 유비쿼터스홈서비스연구팀
{jypark⁰, ehpaik}@etri.re.kr

Design and Implementation of a Mobile Agent System Using OSGi

Jongyoul Park⁰ Euihyun Paik
Ubiquitous Home Service Team, ETRI

요 약

홈네트워크 연구그룹의 표준 중 하나인 OSGi는 OSGi Alliance에서 정의한 표준 서비스 분배 방식이다. 본 논문에서는 이동 에이전트를 OSGi 기반의 홈네트워크에서 연동하기 위해 OSGi 표준의 번들 라이프 사이클을 수정하여 이동 에이전트 시스템은 지원하는 방법을 제안한다. 이동 에이전트는 뛰어난 이식성과 효율적인 관리의 장점에도 불구하고 보안의 문제로 사용이 제한되어 왔다. 홈네트워크는 서비스 분배에서 가입자와 서비스 분배자 사이의 폐쇄적인 구조를 가지고 있기 때문에 이러한 보안상의 취약점은 해결할 수 있다. 따라서 홈네트워크 환경에서 이동 에이전트의 지원은 보다 효율적이고 지능화된 서비스를 가능하게 할 것이다. OSGi의 구조는 서비스를 분배하고 수행하기 위한 번들 라이프사이클을 제공한다. 별도의 에이전트 서버 번들을 구현하는 대신에 본 논문에서는 번들 라이프사이클에서 이동 에이전트가 직접 수행될 수 있는 번들 라이프 사이클을 제안한다.

1. 서 론

홈네트워크는 사용자의 편의성을 극대화한 시스템으로 맥내에 산재되어 있는 가정 기기들을 하나의 네트워크로 묶는 것을 의미한다. 홈네트워크를 구성하는 방법은 크게 유선을 이용하는 방식, 무선선을 이용하는 방식 그리고 유, 무선 통합 방식이 연구되고 있다. 유선을 이용한 방식은 주로 가전 업계를 중심으로 PLC(power line comm.) 방식을 활용하여 가정에서 리모컨으로 전자제품을 다루듯이 쉽게 사용할 수 있는 방법을 연구 중에 있다. 무선선을 이용한 방식은 WPAN, ZigBee, RFID, 무선 IEEE 1394 중심으로 가정내 기기들을 무선으로 묶는 방법을 연구 중에 있다. 유,무선 통합 방식은 해당 기기의 특징을 고려하여 필요한 유,무선 방식을 취하고 무선에 연결된 기기와 유선에 연결된 기기들 사이에 연결(bridge) 기능을 제공한다.

홈네트워크 관련된 표준으로 DLNA, UPnP, PLC, Jini, HAVi, OSGi[1] 등의 다양한 방식들이 추진 중에 있다. 그 중에서 OSGi는 개방형의 특징을 표방한다. 특정 시스템이나 프로토콜에 종속되지 않는 구조를 가지고 있는 것이 특징이다.

홈네트워크는 통신기술과 지능형 서비스로 대표되는 서

비스이다. 지능형 서비스는 에이전트 기술로 대표되며, 현재까지는 멀티에이전트 시스템을 도입하는 경우가 많았다 [7]. 멀티에이전트 시스템은 각각의 서버에 독립된 에이전트가 위치하고 서로의 통신 및 업무를 협동하는 구조를 가지고 있다. 하지만 OSGi와 같은 개방형 구조에서는 서비스의 분배기능을 위해서 OSGi 구조 속에 라이프사이클을 지원하게 된다. 실제 에이전트가 가지는 라이프 사이클은 이 OSGi의 번들 라이프 사이클과 상당히 비슷하다.

이동 에이전트 기술은 지능형 기능 이외에 자율적인 코드 이동성을 지원하기 때문에 많은 연구가 진행되었지만, 보안상의 문제 때문에 널리 사용되지 못했다. 서버 입장에서 이동 에이전트 코드는 외부의 악의적인 코드와 효과적으로 구별할 방법이 없었기 때문이다. 하지만 OSGi 기반의 홈네트워크 시스템은 새로운 서비스를 분배할 수 있는 구조를 가지고 있으며 그 구조가 비교적 폐쇄적(맥내 가족 구성원 과 가정기기들만의 네트워크)이기 때문에 보안 기능의 구현이 가능하다.

그림 1은 OSGi의 서비스 분배 과정이 특정 프로토콜에 종속되어 않는다는 것을 보이고 있다. 각각에 OSGi가 설

치된 서버에 중앙의 관리 시스템에 의해서 전송되고 수행되는 방식을 취하고 있다.

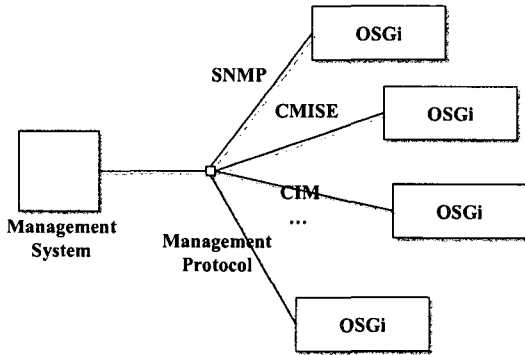


그림 1 서비스 분배 프로토콜

관리 시스템이 각각의 “OSGi 서버에 새로운 번들을 제공하는 것”은 “이동 에이전트의 코드를 전송하는 것”과 유사한 구조를 가지고 있다. 반면 현재 개발된 시스템은 OSGi 구조 위에 올라가지 못하고 별도의 시스템에 연동되는 수준에 머무르고 있다[7]. 본 논문에서는 OSGi 내부에서 이동 에이전트를 지원하기 위해 수정된 라이프 사이클의 정의하고 이동 에이전트 시스템을 구현했다.

2. OSGi 기반의 서비스 기술

그림 2는 OSGi의 서비스 구조를 보이고 있다. 크게 라이프사이클 관리, 레지스트리 관리, 번들 관리, 보안 모듈로 구성이 된다. 특히 “클래스 로딩 + 라이프사이클”은 외부의 코드를 수용하기 위해 설계된 모듈로 에이전트 시스템과 동일한 구조를 가지게 된다.

클래스로딩(ClassLoading)은 자바 Class 파일을 메모리로 상주시키고 새로운 인스턴스를 수행 시키는 기능을 의미하며, 라이프사이클 관리는 에이전트의 생성에서 소멸까지 상태 정보를 관리하는 모듈이다. 현재의 이동 에이전트 기술로는 상태정보를 완전히 유지하면서 이동하는 시스템이 존재하지 않는다. 이동 에이전트가 프로그램의 수행정보를 가지고 이동 하기 위해서는 운영체제에서 프로그램 스택과 수행 중에 연결된 디바이스들을 동일하게 할당해야 하지만 그것이 현실적으로 어렵다. 따라서 데이터에 상태정보를 저장하고 다른 호스트로 이동하는 경우 상태 정보를 복원하는 라이프사이클 방식을 취하고 있다.

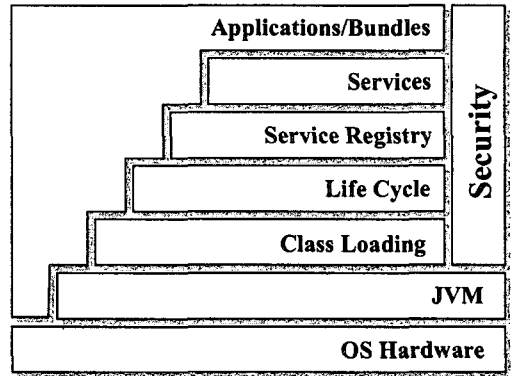


그림 2 OSGi 시스템 서비스 구조

OSGi에서 클래스로딩과 라이프사이클은 외부에서 전송된 번들을 수행시키기 위해서 설계 되었다. 그림 3은 OSGi의 라이프 사이클을 보이고 있다.

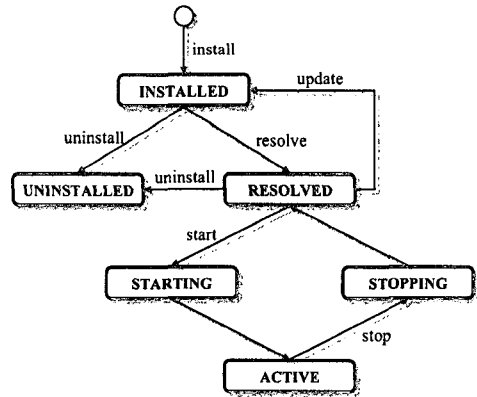


그림 3 OSGi 라이프 사이클

이동 에이전트란 한 서버에서 다른 서버로 코드가 직접 이동하여 수행하는 것을 의미한다. 기존의 에이전트 기술에 이동성을 지원하기 위한 추가적인 라이프 사이클을 가지고 있다. 기본적인 생성, 준비, 수행, 정지, 소멸 이외에 이동 상태가 추가적으로 존재한다. 이동 상태는 현재의 서버에서 에이전트와 관련된 데이터를 삭제하고 지정된 서버로 코드를 이동하는 일련의 과정을 포함한다. JAVA에서 이동성을 지원하기 위해서는 RMI와 Object Serialization을 통해서 원격지에 이동 에이전트 코드를 전송하는 방식을 사용하고 있다[6]. 3장에서 OSGi의 기존 라이프사이클에 영향을 주지 않고 이동 에이전트의 이동성을 지원

하는 새로운 라이프 사이클을 정의하고 실제 이동 에이전트 시스템에 적용하여 구현 결과를 보인다.

3. OSGi 기반의 이동 에이전트 서버의 설계 및 구현

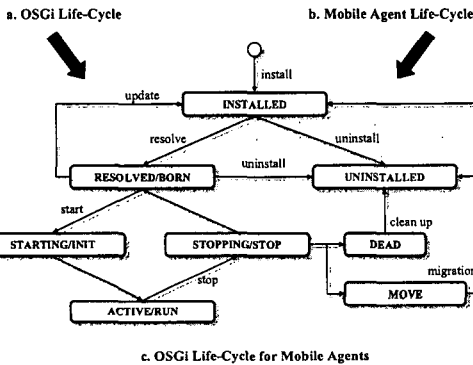
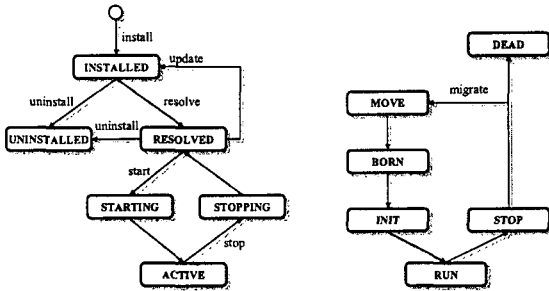


그림 4 이동 에이전트를 지원하는 라이프사이클

그림 4-a,b 는 OSGi와 이동 에이전트의 라이프 사이클을 보이고 있다. 두 시스템은 서로 다른 구조를 가지고 있지만, 기본 상태인 RESOLVED(BORN), STARTING(INIT), ACTIVE(RUN), STOPPING(STOP) 부분이 공유되고 있는 것을 알 수 있다. 그림 4-c는 두가지 라이프 사이클을 모두 지원하는 상태 정보를 도식한 것이다. 기본적인 4가지 상태 이외에 DEAD, MOVE 두가지 상태가 추가된다. DEAD는 에이전트가 소멸하는 경우로 서비스가 UNINSTALL 되는 것과는 다른 의미를 가진다. UNINSTALL 은 서비스를 종료하는 것이지만 DEAD는 사용자에게 최종 보고를 하거나 에이전트의 정보를 검증하는 과정이 추가 되기 때문에 별도로 존재한다. MOVE는 이동 에이전트가 이동 하는 과정으로 수행 서버의 데이터를 정리하고 새로운 서버로의 이동을 위한 작업을 수행하는 과정이다. MOVE를 수행한 후에 서버는 UNINSTALLED 상태로 변경되고 에이전트가 이동한 서버는 INSTALLED 상태로 변경된다.

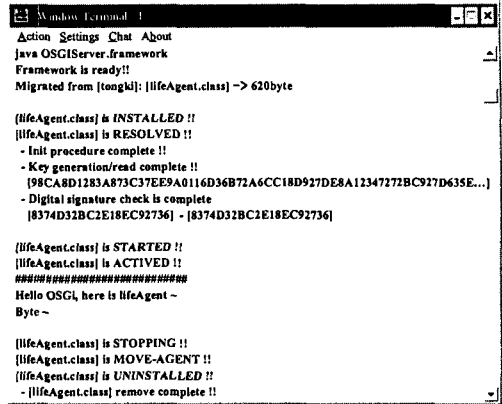


그림 5 이동 에이전트와 연동 화면

그림 5는 실제 에이전트 시스템인 X-MAS[6]에서 수정된 OSGi의 라이프 사이클로 이동 에이전트를 수행한 결과 화면을 보이고 있다.

4. 결 론

홈네트워크에서 에이전트 기술의 도입은 필수적이다. 기존의 연구는 단순 에이전트 시스템과의 연동(별도 시스템) 수준에 그치고 있으나 OSGi의 라이프 사이클을 수정하여 특수한 서비스 번들로 구동이 가능하기 때문에 홈네트워크의 구성이 간략화 되고 효과적인 연동이 가능하다. 향후 이동 에이전트를 이용하여 맥내의 OSGi 번들을 관리하는 기능에 대한 연구가 필요하다.

참고문헌

1. OSGi Specification v3.0, www.osgi.org, 2004
2. D. Zhang, H. Lee, X. Ni, S. Zheng, "Open Service Residential Gateway For Smart Homes", 2003.
3. Li Gong, "A Software Architecture for Open Services Gateways," IEEE Internet Computing, Jan. 2001, pp 64-70.
4. D. Valtchev and I. Frankov, "Service Gateway Architecture for a Smart Home," IEEE Commun. Mag., April 2002, pp 126-132.
5. K. Hofrichter, "The Residential Gateway as Service Platform," IEEE International Conf. On Consumer Electronics, USA, 2001, pp 304-305.
6. J. Park, D. Lee, H. Lee, "Data Protection in Mobile Agents: one-time key based approach," IEEE ISADS 2001, USA, 2001, pp 411-418.
7. H. Lee, K. Kang, J. Lee, "A Study for FIPA-OS Multi-Agent Framework in OSGi Service Platform," International Symposium on Advanced Intelligent Systems, 2003, pp 232-235.