

# 규칙 기반의 홈 서버 아키텍처 설계

황도현<sup>0</sup> 이현정 정기원

송실대학교 대학원 컴퓨터학과

hankee@nate.com<sup>0</sup>, kiwi@empal.com, chong@computing.ssu.ac.kr

## Design for Rule-based Home Server Architecture

Dohyun Hwang<sup>0</sup>, Hyunjeong Lee, Kiwon Chong  
Dept. of Computing, Graduate School, Soongsil University

### 요 약

본 연구는 미들웨어 기반의 OSGi(Open Service Gateway initiative) 서비스 플랫폼과 규칙 기반 생성 시스템 기술을 응용하여 유연하고 효율적인 홈 서버 아키텍처를 제안한다. 미들웨어 컴포넌트인 서비스 룰 엔진은 서비스로부터 요청된 호출을 정해진 규칙에 따라 평가, 판단하여 홈 기기들을 제어하고 관리한다. 서비스 룰 엔진은 실행 규칙이 명세 되어 있는 프로파일을 참조하여 처리하고, 홈 기기 접근 권한과 현재 서로 상충되는 홈 기기가 작동 중인지 점검하며, 우선순위와 충돌 방지 조건 및 제약조건을 조회하여 홈 기기를 제어한다. 이때 방재, 보안에 관련된 홈 기기의 작동이 요구되는 경우 정해진 방재, 보안 규칙에 따라 가장 먼저 수행된다. 프로파일은 사용자가 서비스 룰 매니저를 통해 관리하거나 외부의 접근 허가된 사용자가 관리할 수 있으며, 여러 형태의 프로파일이 제공되어 사용자의 상황에 따라 프로파일을 선택하여 홈 서버에 적용할 수 있다.

### 1. 서론

홈 네트워크는 최근 널리 퍼지고 있는 유비쿼터스 컴퓨팅 환경과 함께 인간의 거주 환경에 매우 큰 영향을 미치고 있으며 모든 공학 분야와 인문 분야의 주요 연구 대상이 되고 있다.

현재 홈 네트워크 기술의 도입은 주로 하드웨어와 통신 분야 위주로 이루어지고 있어 균형적인 발전을 위해 소프트웨어 기술에 보다 많은 관심과 노력이 필요하다. 거주환경은 점차 정보화, 지능화 되어 가고 있으며 이에 따른 소프트웨어의 발전도 기대되고 있다.

홈 네트워크 소프트웨어 기술은 하드웨어의 기능이 정확하고 신속하게 수행될 수 있도록 지원해야 하며 사용자가 직접적으로 홈 기기와 관련되어 있는 홈 네트워크의 특성상 안전성과 편의성이 가장 중요하다. 또한 홈 네트워크 사용자의 요구사항은 주거의 형태, 거주자의 라이프 스타일(Life Style)에 따라 매우 다양하기 때문에 이러한 요구를 충족할 수 있는 소프트웨어가 제공되어야 한다.

그러나 홈 네트워크에서 단순 제어 위주의 소프트웨어는 다양한 고객의 요구를 만족시킬 수 없으며 안전성과 편의성을 위한 서비스의 제공을 불가능하게 한다. 홈 네트워크 제품의 안전성과 편의성을 제공하고 고부가가치를 위해 각종 다양한 서비스를 지원할 수 있는 소프트웨어의 개발이야말로 매우 중요하다고 할 것이다.

이러한 기술을 위해 플랫폼에 독립적이며 이 기종간 호환을 가능하게 하는 미들웨어(Middleware)기술과 다양한 서비스 및 요구의 충족시킬 수 있는 규칙 기반 기술을 도입하는 것은 이러한 문제를 해결할 열쇠이다. 이를 바탕으로 각 하드웨어 간 우선순위, 충돌방지, 상태에 따른 제어 규칙, 제약 조건, 방재, 보안 처리 등을 관리할 수 있는 소프트웨어 기술이 반드시 필요할 것이다.

본 연구는 이를 위한 관련연구로 홈 네트워크 아키텍처 기술, 플랫폼 독립적이고 이기종간 호환을 가능하게 하는 OSGi(Open Service Gateway initiative) 서비스 플랫폼(Service Platform), 그리고 규칙 기반(Rule Based) 생성 시스템(Production System)을 설명하고 있으며 홈 네트워크 소프트웨어 핵심 기술로 규칙 기반(Rule Based)의 홈 서버 기술을 제안하고자 한다.

### 2. 관련연구

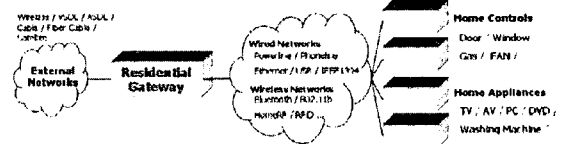
#### 2.1. 홈 네트워크 아키텍처

홈 네트워크는 대내에 홈 기기(Home Device)들이 유무선 네트워크로 연결되어 원격 또는 조정기에서 제어되는 시스템이다. 또한 홈 오토메이션(Home Automation), 디지털 홈(Digital Home), 스마트 홈(Smart Home)등으로 명명되기도 한다.

홈 기기는 홈 가전(Home Appliance)과 홈 제어기(Home Control)로 구분할 수 있다. 홈 가전은 TV, 전화, 오디오, PC, 비디오, DVD, 냉장고, 세탁기, 에어컨, 가스오븐 등 사용자에게 의해 개별적으로 제어 가능한 기기들이고 홈 제어기는 건축물 내에 포함된 환경, 설비 구성요소인 조명, 창문, 문, 환기, 실내온도, 방재, 보안, 가스, 상하수도 등을 제어하기 위한 기기들이다.

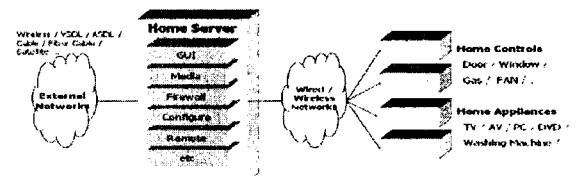
홈 네트워크 아키텍처로는 거주형 게이트웨이(Residential Gateway)형과 홈 서버(Home Server)형이 있다.

거주형 게이트웨이형은 외부에서 인터넷 등과 같은 네트워크망을 통해 대내의 네트워크로 접근하여 대내의 기기들을 제어, 관리하는 형태이다. 게이트웨이는 외부와 내부 네트워크를 연결시켜주는 관문 역할을 주로 수행하며 비교적 홈 기기 관리 등의 서비스의 수행이 적은 형태이다[1][2][3].



[그림 1] 거주 게이트웨이(Residential Gateway)형

홈 서버형은 홈 서버에서 여러 형태의 서비스를 지원하는 형태이다. 대내 기기들을 통합 관리하고 네트워크, 보안, 원격관리, JVM, Web, FTP, DHCP 서비스 등의 기능을 홈 서버에서 수행한다. 네트워크의 게이트웨이 역할과 함께 각종 서비스를 함께 제공하는 서버로의 역할을 수행한다[4][5].



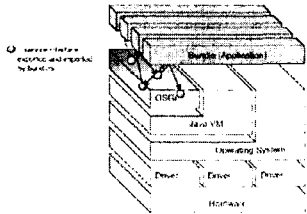
[그림 2] 홈 서버(Home Server)형

2.2. OSGi(Open Service Gateway initiative)의 서비스 플랫폼

OSGi는 홈 네트워크, 텔레메틱스, 산업 자동화, 기타 정보 가진 및 다양한 단말 환경에 서비스를 배포, 관리할 수 있는 개방형 서비스 게이트웨이 소프트웨어 아키텍처를 개발하고 표준화 시키기 위해 세계적인 업계들에 의해 설립된 국제표준화 단체이다.

OSGi 서비스 플랫폼은 게이트웨이 상의 이 기종 기술 또는 타사의 서비스 간에도 통신이 가능하게 하는 표준 기술을 미들웨어 형태로 제공한다. 이는 새로운 기술이나 서비스를 제공하는 장치가 추가 되더라도 장치 설치자의 요구 또는 거주자에 의한 간섭 없이도 인터넷을 통해 새로운 소프트웨어 모듈을 손쉽게 설치 및 업그레이드, 교환 시키는 것을 가능하게 한다[6][7][8][9].

OSGi 서비스 플랫폼의 아키텍처는 OS(Operating System) 위에 JVM(Java Virtual Machine)이 기본적으로 존재하고 그 위에 번들의 관리와 실행환경을 담당하는 OSGi 프레임워크(Framework)가 미들웨어 형태로 존재하고, 그 위에 여러 서비스의 수행을 담당하는 번들(Bundle 또는 Application)이 JAR 형태로 구성되어 있다.



[그림 3] OSGi 서비스 플랫폼 아키텍처 [13]

OSGi 서비스 플랫폼의 큰 특징은 다음과 같다

- 유연한 소프트웨어 컴포넌트 모델
- 개방형 프로토콜과 표준
- 소프트웨어 모듈 재사용
- 동적 소프트웨어 업데이트
- 원격 관리, 유지보수, 진단
- 보안, 인증 규칙
- 플랫폼 독립성

OSGi 서비스 플랫폼은 다음과 같은 서비스를 지원한다.

- 프레임워크 서비스(Framework Services) - Permission Admin, Package Admin, Start Level, URL Handlers
- 시스템 서비스(System Services) - Log Service, Configuration Admin, Device Access, User Admin, IO Connector, Preferences Service
- 프로토콜 서비스(Protocol Services) - Http Service, UPnP
- 기타 서비스(Miscellaneous Services) - Wire Admin, XML Parser

이러한 미들웨어 형태의 OSGi 서비스 플랫폼은 유연하고 개방적인 구조를 가진다. 본 연구는 이러한 구조를 바탕으로 다양한 홈 기기의 제어 및 관리를 담당하는 부분과 서비스의 수행 규칙을 담당하는 부분을 분리함으로써 각 기능을 정확하고 효율적으로 수행하는 소프트웨어 컴포넌트의 제공을 가능하게 한다.

2.2 규칙 기반(Rule-based)의 생성 시스템(Production System)

규칙 기반 시스템(Rule-based System)을 기반으로 한 생성 시스템(Production System)은 'IF Conditions(조건) THEN Actions(행동)' 형식의 추론 규칙으로부터 문제를 해결하는 것으로, 조건(Conditions) 부분에 일정한 생성 규칙(Production Rule)을 기술하고 행동(Actions) 부분에 조건이 만족되었을 때 취할 적절한 행동을 기술한다. 이를 바탕으로 문제 해결을 위한 추론 엔진(Inference Engine)을 두어 주어진 문제에 대하여 이들 규칙(Rule)을 차례차례로 적용함으로써 해당(결론)을 도출한다.

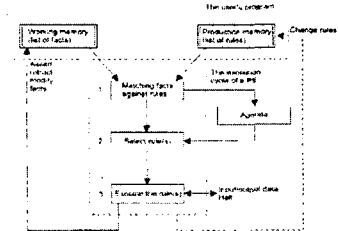
생성 시스템(Production System)은 크게 3 부분으로 구성된다.

- 생성 규칙들(Production Rule)의 모임인 생성 메모리(Production Memory)
- 현재의 상태를 나타내는 버퍼(Buffer)와 비슷한 데이터 구

조로써의 작업 메모리 (Working Memory 또는 List of Facts, Context)

- 작업 메모리와 생성 메모리를 근거로 문제 해결을 위한 추론 엔진(Inference Engine, 또는 Interpreter)

다음은 생성 시스템의 실행을 나타내는 그림이다.



[그림 4] 생성 시스템 실행 주기(A Production System Cycle) [12]

생성 시스템의 수행 중 동일 조건에 대한 규칙이 여러 개일 수 있다. 이러한 규칙들의 집합을 충돌집합(Conflict Set)이라고 하며, 시스템은 이들 중 하나 또는 경우에 따라 그 이상을 선택하여야 한다. 이를 위하여 다음을 포함한 여러 정책이 제시되어 왔다.

- ① 작업 메모리에 의하여 만족되는 최초의 규칙 : 여기서 최 초라 함은 규칙이 생성 메모리에 놓인 순서를 말할 때 제일 앞에 놓인 것이다.
- ② 가장 높은 우선순위(Priority)를 가진 규칙을 선택 : 이를 위하여 프로그래머는 문제 분야의 성격에 따라 각 규칙에 적당한 우선순위를 부여한다.
- ③ 현재의 작업 메모리에 만족되는 가장 특수한(Specific) 규칙, 즉 가장 세분화된 조건을 갖는 규칙. 다시 말하면 조건 부를 구성하는 논리곱요소(Conjunct)가 가장 많은 규칙.
- ④ 작업 메모리에 삽입된 가장 최근의 정보에 만족되는 규칙.
- ⑤ 가장 새로운 규칙, 즉 이전에 수행되지 않았던, 혹은 수행되었더라도 다른 변수값으로 수행되었던 규칙.
- ⑥ 임의의 규칙.
- ⑦ 순서에 상관없이 만족되는 모든 규칙을 병행하여 수행.

여러 다른 생성 시스템은 이렇게 간단한 충돌 해결 방법 중의 서로 다른 조합을 사용하고, 어떤 것은 아주 복잡한 스케줄링 알고리즘(Scheduling Algorithm)을 사용하기도 한다[10][11][12].

본 연구에서 제시되는 서비스 규칙 엔진은 특정 조건에서 가장 최선의 선택을 위한 문제 해결 방식이 적용되며 그 일환으로 다양한 충돌 해결 방법이 선택된다.

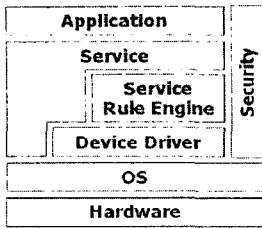
3. 규칙 기반(Rule-based)의 홈 서버 연구 방안

앞의 관련연구에서 제시된 미들웨어 기반의 OSGi 서비스 플랫폼과 규칙 기반 생성 시스템 기술의 응용은 홈 네트워크에 있어서 유연하고 효율적인 서비스를 제공할 수 있다.

일반적인 홈 기기들은 특정 상태, 조건, 시간에 의해 개별적으로 제어된다. 이러한 기능의 수행은 동일 조건에 따라 각 기기들이 서로 상충되는 작동을 함으로서 자칫하면 큰 위험을 초래할 수 있다. 예를 들어 환기 시설과 화재 방지 시설의 경우가 있다. 화재가 발생하여 실내의 온도가 상승하였을 때 환기 시설은 이 경우 온도의 상승으로 판단하여 실내에 새로운 찬 공기(산소)를 유입할 것이고, 화재 방지 시설은 실내에 더 큰 화재의 방지를 위해 새로운 공기(산소) 차단을 하려 할 것이다. 이러한 경우 기기는 오작동을 일으켜 거주자에게 큰 위험을 초래할 것이다. 이뿐만 아니라 방법 시설과 창문-문 개폐 시설, 각기 다른 거주자에 의한 동일 기기 동시 제어 등의 경우가 있다.

본 연구에서 제시하는 규칙 기반의 홈 서버 아키텍처에서는 이러한 안전과 편의를 위협하는 경우를 없애기 위해 각 서비스들이 홈 기기를 직접 제어하는 것이 아니라 서비스를 엔진이라는 미들웨어 컴포넌트를 통해 제어한다. 이러한 규칙 기반의 서비스를 엔진(Service Rule Engine)을 도입한 소프트웨어 아키텍처를 제시함으로써 안전성과 편의성, 사용자의 다양한 요구를 충족시키고자 한다.

규칙 기반의 홈 서버 소프트웨어 아키텍처는 다음과 같다.



[그림 5] 규칙 기반의 홈 서버 소프트웨어 아키텍처

기본적인 소프트웨어 아키텍처는 OS, 디바이스 드라이버(Device Driver), 서비스 룰 엔진(Service Rule Engine), 서비스(Service), 어플리케이션(Application), 보안(Security)로 구성되어 있다. 어플리케이션은 홈 기기의 제어 및 관리를 요청하기 위한 소프트웨어로 사용자와 서비스 사이의 인터페이스이다. 이 어플리케이션은 사용자가 직접 호출하거나 외부 네트워크에서 원격으로 호출 가능하다. 서비스는 기본적으로 안전성 있는 홈 기기 제어를 위한 서비스 룰 엔진 호출, 서비스 접근 권한 관리, 드라이버 및 서비스 동적 업데이트, 일반적 서비스(Web, DHCP, 데이터베이스, 멀티미디어)로 구성된다.

서비스 룰 엔진은 서비스로부터 요청된 호출을 정해진 규칙에 따라 평가, 판단하여 홈 기기들을 제어하고 관리하는 미들웨어 컴포넌트이다. 일련의 상태, 조건, 시간이 입력되면 서비스 실행 우선순위, 충돌 방지, 상태에 따른 제약조건, 방재, 보안 처리, 접근 권한, 서비스 실행의 규칙을 위한 프로파일(Profile, 규칙명세) 규칙에 따라 결과를 선택하여 수행한다.

서비스가 서비스 룰 엔진을 호출하면 기본적인 홈 기기의 제어는 실행 규칙이 명세 되어 있는 프로파일에 의해 처리 된다. 엔진은 홈 기기 접근 권한과 현재 서로 상충되는 홈 기기가 작동 중인지 점검하고 우선순위와 충돌 방지 조건 및 제약조건을 조회하여 홈 기기를 제어한다. 이때 방재, 보안에 관련된 홈 기기의 작동이 요구되는 경우 정해진 방재, 보안 규칙에 따라 가장 먼저 수행된다. 이러한 규칙을 정의(신규, 수정, 삭제)하기 위해 사용자는 서비스 룰 매니저(Service Rule Manager)를 통해 관리할 수 있다.

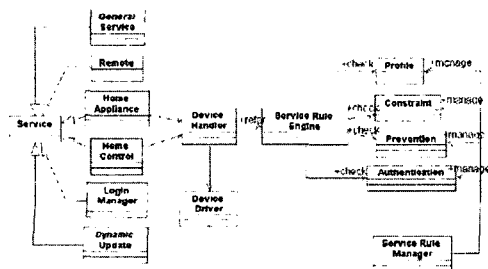
프로파일은 사용자가 서비스 룰 매니저를 통해 관리하거나 외부의 접근 허가된 사용자에 의해 관리 될 수 있으며 여러 형태의 프로파일이 제공되어 사용자의 성장(수험생용 프로파일, 맞벌이부부용 프로파일 등)에 따라 프로파일을 선택하여 홈 서버에 적용할 수 있다.

디바이스 드라이버는 홈 가전과 홈 제어기의 센서와 구동기기의 작동을 위한 소프트웨어로 서비스나 서비스 룰 엔진에 의해 호출되어 사용된다.

OS는 홈 서버를 기본적으로 운영하기 위한 CPU, 메모리, 디스크, 입출력 등의 자원을 관리하고 네트워크에 관련된 기능을 수행한다. 보안은 소프트웨어 보안을 위해 아키텍처 전반에 걸쳐 제공이 된다.

4. 서비스 규칙 엔진 소프트웨어 구현

다음은 아키텍처의 주 핵심요소인 서비스와 서비스 룰 엔진의 클래스 다이어그램이다.



[그림 6] 서비스 룰 엔진 클래스 다이어그램

서비스 관련 클래스의 간략한 설명 다음과 같다.

- General Service - 서버의 일반적 기능 수행을 위한 클래스
- Remote - 원격 검침 및 관리를 위한 클래스
- Home Appliance - 홈 가전 제어를 위한 클래스
- Home Control - 홈 제어(창문, 문, 환기, 환경적요소)를 위한 클래스
- Login Manager - 서비스 접근 제어를 위한 클래스
- Dynamic Update - 드라이버 및 소프트웨어의 동적 업데이트를 위한 클래스

서비스 룰 엔진관련 클래스는 간략한 설명은 다음과 같다.

- Device Handler - 홈 드라이버를 제어하는 클래스
- Service Rule Engine - 서비스 룰을 제어하는 클래스
- Profile - 서비스 제어 규칙 명세를 위한 클래스
- Constraint - 제약조건을 위한 클래스
- Prevention - 방재, 보안을 위한 클래스
- Authentication - 홈 기기 접근 제어를 위한 클래스
- Service Rule Manager - 프로파일, 제약조건, 방재, 접근제어 정보를 관리하는 클래스

서비스에서 홈 가전과 홈 제어를 관리하는 Home Appliance, Home Control 클래스는 Device Handler 클래스를 통하여 Service Rule Engine 클래스를 호출한다. Service Rule Engine 클래스는 Profile, Constraint, Prevention, Authentication 클래스를 호출하여 서비스 룰을 점검한 후 Device Handler 클래스가 Device Driver를 호출하여 홈 기기를 제어한다.

5. 결론

본 연구에서는 하드웨어 간 우선순위, 충돌방지, 상태에 따른 제어 규칙, 제약 조건, 방재, 보안 처리, 접근제어 등을 관리할 수 있는 서비스 룰 엔진을 적용한 미들웨어 컴포넌트 소프트웨어 기술을 제시하였다. 이러한 기술은 점차 발전하고 있는 홈 네트워크에 위협 요소로 작용하는 홈 기기간 충돌, 제어 불능 등의 문제를 보완하는 안전성 있는 서비스를 지원 할 수 있으며 주거의 형태, 거주자의 라이프 스타일(Life Style) 변화에 따른 다양한 요구사항의 변화를 충족시킬 수 있다.

미들웨어 컴포넌트 형태로 규칙 기반(Rule-based) 서비스를 제공하는 규칙 기반의 홈 서버는 보다 안전하고 편리한 서비스 제공을 위한 홈 네트워크 소프트웨어 기술이며 차별화 되고 우수한 품질의 홈 네트워크 서비스를 제공한다.

앞으로의 홈 서버 소프트웨어 기술은 규칙 기반 기술을 바탕으로 보다 학습화, 지능화(Intelligent)될 것이다. 본 연구는 지속적으로 이러한 학습 능력을 가진 적응형 소프트웨어 홈 서버 형태로 발전 할 것이며 최종적으로 그 구현을 그 목적으로 할 것이다.

6. 참고문헌

[1]Corcoran P.M., Mapping Home-Network Appliances to TCP/IP Sockets Using a Three-Tiered Home Gateway Architecture, Consumer Electronics, IEEE Transactions on, Volume: 44, Issue: 3, Pages:729 - 736, Aug. 1998  
 [2]Gong, L., A Software Architecture for Open Service Gateways, Internet Computing, IEEE, Volume: 5, Issue: 1, Pages:64 - 70, Jan.-Feb. 2001  
 [3]http://www.ida.gov.sg/idaweb/techdev/infopage.jsp?infopagecategory=13:techdev &versionid=2&infopageid=12078  
 [4]Changseok Bae; Jinho Yoo; Kyuchang Kang; Yoonsik Choe; Junwoo Lee; Home Server for Home Digital Service Environments, Consumer Electronics, IEEE Transactions on, Volume: 49, Issue: 4, Pages:1129 - 1135, Nov. 2003  
 [5]Saito, T.; Tomoda, I.; Takabatake, Y.; Arni, J.; Teramoto, K.; Home Gateway Architecture And Its Implementation, Consumer Electronics, IEEE Transactions on, Volume: 46, Issue: 4, Pages:1161 - 1166, Nov. 2000  
 [6>About the OSGI Service Platform Technical Whitepaper(v3.0), OSGI Alliance, 2004.07.  
 [7]Valtchev, D.; Frankov, I.; Service Gateway Architecture for a Smart Home, Communications Magazine, IEEE, Volume: 40, Issue: 4, Pages:126 - 132, April 2002  
 [8]http://www.osgi.org  
 [9]http://www.seri.org/fr/IPdsV.html  
 [10]http://www.aistudy.com/expert/rule\_kim.htm  
 [11]Frederick Hayes-Roth, Rule-based systems, Communications of the ACM, Volume 28 Issue 9, Sep 1985.  
 [12]Nikola K. Kasabov, Foundations of Neural Networks Fuzzy Systems and Knowledge Engineering, pp118-132, 1996  
 [13]http://www.javacommunity.org/test\_jco/oldconference/2th/2th\_DATA/conf2-OSGI\_tutorial.pdf