

이질적인 홈 네트워크 미들웨어 상호 연동성 지원을 위한
사용자 중심의 시나리오 기반 통합 구조 설계¹

김민찬^o 김성조
중앙대학교 컴퓨터 공학부
{barrios, sjkim}@konan.cse.cau.ac.kr

Design of a User Oriented Scenario-Based Integrated Architecture for
Supporting Interoperability among Heterogeneous Home Network
Middlewares

Min Chan Kim^{*} Sung Jo Kim
School of Computer Science and Engineering, Chung-Ang University

요 약

정보가전들을 제어하기 위한 많은 미들웨어들이 존재하고 있으며(예 : Havi, Jini, LonWorks, UPnP), 또한 계속 진화하고 있다. 이에 따라 다양한 정보가전들에 대해 특화된 새로운 미들웨어들은 계속해서 등장하게 될 것이다. 또한 비즈니스 시장에서 주로 사용되어오던 웹 서비스가 점차 홈 내부의 서비스로 확대됨에 따라 맥 내의 가전들과 맥 외의 웹 서비스를 상호 연동하기 위한 미들웨어가 필요하게 되었다. 본 논문은 이질적인 홈 네트워크 미들웨어 상호 연동을 위한 통합 구조 방식에 대해 고찰하고, 이질적인 홈 네트워크 미들웨어들을 통합하여 정보 가전들을 제어 및 연동하고, 진화하고 있는 홈 네트워크 미들웨어들의 유동적인 특성을 반영할 수 있는 사용자 중심의 시나리오 기반 통합 구조의 설계에 관하여 설명한다.

1. 서 론

초고속 통신 기술의 발전이 가속화되며, 인터넷의 급속한 발전과 하드웨어 디지털 기술의 발전은 가전들의 지능화를 촉진시키고 있다. 이와 함께 가정내의 장치들을 사용자의 개입없이 제어 및 관리하기 위한 많은 미들웨어들이 등장하게 되었으며 점차 발전하고 있다. 이에 대표적인 미들웨어들로서 UPnP[1], Jini[2], Havi[3], LonWorks[4]이 있으며, 향후에도 특정 분야에 특화된 미들웨어들은 계속해서 등장하게 될 것이다[5][6]. 이러한 미들웨어들은 장치 제어라는 공통의 목표를 가지고 있지만, 각기 다른 구조를 가지고 있다. 각 미들웨어마다 서비스 발견 과정, 서비스를 전달하는 방식이 상이하기 때문에 서로간의 미들웨어 연동이 불가능한 상황이다. 가정 내에 이기종 미들웨어들이 공존하며 연동 될 수 없다는 것은 서비스를 개발하는 개발자에게는 각 미들웨어 지원을 위한 같은 서비스를 중복해서 개발하여야 하는 부담감과 사용자에게는 같은 미들웨어를 사용하는 장치들만이 제어가 가능하기 때문에 홈 네트워크의 활용도를 떨어뜨리게 되고, 이는 곧 홈 네트워크 기술 발전과 시장 성장의 큰 저해 요인으로 작용하게 된다. 또한 홈 네트워크 서비스와 같이 진행되어야 할 연구로서 맥내 가전의 원격 제어가 있다. 맥내 가전의 원격 제어를 위한 다양한 방안이 제안되고 있지만, 비즈니스 서비스 분야에서 주로 발전하고 개발되어 오던 웹 서비스[7]가 차츰 일반적인 서비스로 시장을 확대해가며, 가정 내의 가전들을 제어하고 외부의 서비스들을 연결시켜주는 기술로서 가장 적합한 방식으로 떠오르고 있다. 이에 본 논문은 이기종간의 미들웨어 연동을 지원하며, 원격에서도

제어가 가능한 통합형 미들웨어 개발을 위한 문제점과 그 문제점을 해결하기 위한 새로운 구조를 제안한다. 본 논문의 구성은 2장 관련 연구에서 홈 네트워크 미들웨어 통합 기술과 문제점들에 대해 고찰하며, 3장에서 논문에서 제시하는 통합 미들웨어 구조에 관해 설명하고 4장에서 결론 및 향후 과제에 대해 언급한다.

2. 관련 연구

현재 이기종 미들웨어의 연동 방법에 대한 많은 연구가 진행되고 있다. 미들웨어 연동방식은 크게 2가지로 나누어 볼 수 있다.

2.1 개별 브릿지 방식의 미들웨어 통합

이기종 미들웨어들을 연동하기 위한 방안으로 1:1 미들웨어 브릿지 방식을 이용하여 상호 호환성을 보장하려는 연구가 많이 시도되고 있다. 홈슨 멀티미디어에서는 UPnP-to-Havi 브릿지가, 필립스에서는 Havi-to-UPnP 브릿지가 각각 개발되어 졌다. 또한 New Orleans 대학에서도 Jini와 UPnP를 연동하기 위한 연구가 진행되었다. 하지만 1:1 브릿지 모델의 경우 2가지의 미들웨어 사이에서의 연동은 용이하지만 미들웨어의 수가 늘어나면 브릿지의 숫자도 증가하게 되고 연결 관계가 복잡해질 수 있으며, 다양한 미들웨어들을 연동하기 위한 일관된 방법을 제시하지 못하고 있어 확장성에 문제점이 있다.

2.2 통합 구조 기반의 미들웨어 통합

다양한 미들웨어들의 상위에서 공통 계층을 제공하며 공통 계층을 중심으로 각 미들웨어들을 브릿지하여, 새로운 미들웨어

¹ 본 연구는 정보통신부 ITRC 프로그램의 지원을 받아 중앙대학교 HNRC(Home Network Research Center)에서 수행되었음.

출현에도 쉽게 통합할 수 있다는 장점을 가지고 있다. 왜세다. 대학에서는 SOAP (Simple Object Access Protocol)[8] 게이트웨이를 구성하여 미들웨어 통합을 시도하려는 연구가 있었으며, OSGi Alliance나 국내 ETRI등에서도 이기종 미들웨어간 연동 서비스에 대한 연구가 계속해서 진행되고 있다. 하지만 하나의 인터페이스에 여러 미들웨어들의 인터페이스를 브릿지하여 연동하는 방식은 몇가지 문제점이 존재한다. 이러한 문제점에 다음절에서 논의한다.

2.3 통합 구조 기반 방식의 문제점

본 논문에서 제시하고 있는 모델 또한 통합 구조 기반 방식의 모델이다. 따라서 통합 구조 기반 방식의 문제점에 대해 자세히 살펴보기로 한다.

첫째, 어떻게 서로 다른 미들웨어의 장치를 투명하게 발견할 것인가? 각 미들웨어들은 서로 다른 서비스 발견(Service Discovery)방식을 사용하고 있다. 이러한 서비스 발견 방식의 차이는 같은 서비스를 제공하는 장치라도, 장치가 지원하는 미들웨어가 다르다면 서로 인지 할수 없는 문제를 야기하게 된다.

둘째, 어떻게 서로 다른 미들웨어의 서비스를 호출할 것인가? Jini의 서비스 호출 방식은 Java 바이트코드를 이용하여 Java RMI(Remote Method Invoke)를 사용하는 방식이고, UPnP의 경우 SOAP(Simple Object Access Protocol)을 사용하여 텍스트 스트림을 전송하여 호출하는 방식이다. 이러한 서비스 호출 방식의 매커니즘의 차이는 이기종 미들웨어 연동에서 반드시 해결돼야 할 문제이다. 이 문제를 해결하기 위해서는 먼저 미들웨어간의 문법적인 요소(메소드 이름, 인자의 배치 순서, 각 인자나 리턴 값의 타입의 크기)를 보정하고, 각 미들웨어 서비스 호출 매커니즘에 따르는 호출 방식으로 변환이 필요하다.

셋째, 이 기종간의 서비스가 상호 호환 가능하다는 것을 과연 어떻게 알수 있을까? 서비스 인터페이스가 같다면 같은 서비스를 제공하는 서비스로 인지하여야 할 것인가? 예를 들어 보도록한다. Jini의 스토리지 서비스가 제공하는 메소드는 그림 1과 같고 제공하는 서비스는 새로운 데이터를 파일로 저장하여 보관하는 역할을 한다고 가정한다. 마찬가지로 UPnP의 스토리지 서비스는 그림 2와 같은 SCPD(Service Control Protocol Description)안에 PutFile 메소드를 가지고 있다.

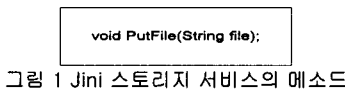


그림 1 Jini 스토리지 서비스의 메소드

```

...
<action>
  <name>PutFile</name>
  <argumentlist>
    <argument>
      <name>file</name>
      <relatedStateVariable>newFile</relatedStateVariable>
      <direction>in</direction>
    </argument>
  </argumentlist>
...
    
```

그림 2 UPnP 스토리지 SCPD

하지만 UPnP 스토리지 서비스는 먼저 사용자의 인증과정을 거쳐야만 하고, 그 과정을 통과한 후 새로운 데이터를 파일로 저장하게 되며, 사용자 인증이 되어있지 않을 경우 에러를 반환한다고 가정한다. 위의 경우 서비스 인터페이스의 문법(syntax)적인 요소는 비슷하지만 의미론적(semantic) 요소의 분명한 차이로 인하여 상호 연동을 하기 위해서는 의미론적 요소의 보정을 위한 방안이 필요하다. 이렇게 문법상으로는 일치하지만

의미상으론 일치하지 않는 서비스를 상호 연동하기 위해서 각 서비스(예: TV, MP3, 프린터 등)마다 각 미들웨어별 테이블 형태의 변환 브릿지를 두는 방식등을 사용할 수 있다. 하지만 이러한 정적인 방식은 새로운 기능이나 장치가 추가될 때마다 변하게 되는 홈 네트워크 미들웨어 프로토콜의 동적인 특성을 반영하기에는 한계를 가지고 있다.

3. HOMI (Homenetwork Middleware for Interoperability) 구조
 HOMI는 본 논문에서 제안하는 통합 구조의 명칭이며, 홈 네트워크 서비스의 사용 형태를 반영하여 설계 되어진 사용자 중심의 시나리오 기반 통합 구조이다. 예는 홈 네트워크의 사용 시나리오를 나타낸다.” 일요일 오전 홍길동씨는 가족들과 함께 놀이공원으로 나들이를 갔다. 홍길동씨는 디지털 카메라를 사용하여 많은 사진들을 찍으며 즐거운 시간을 보내고 집으로 돌아온다. 집에 들어오는 순간 집안의 스토리지 서비스는 카메라에 새로운 사진들을 발견하고 파일 서버에 저장하게 되며, 이 때 집안에 걸려 있는 액자가 반응하여 새로운 사진으로 교체할 것인가를 홍길동씨가 소유하고 있는 PDA를 통하여 질문한다. 홍길동씨는 PDA에 뜬 메시지를 보고, 오늘 찍은 다양한 사진들 가운데 제일 맘에 드는 것을 골라 선택하자 그 새로운 사진이 액자에 나타나게 된다.” 시나리오 속에 등장하는 각 장치들이 서로 다른 이기종 미들웨어를 사용하는 장치라고 가정하였을 때, 결국 최종 사용자 입장의 관점에서 홈 네트워크 서비스는 임의의 시나리오를 가지고 특정 사건을 중심으로 다양한 장치들이 연동되는 방식이 된다. 본 논문에서는 사용자들의 홈 네트워크 서비스 사용의 편의성을 제공하며, 통합 구조 기반 방식의 문제점들을 해결하기 위하여 그림 3과 같은 구조를 제안한다.

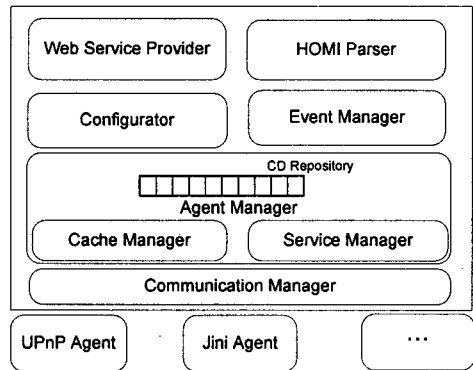


그림 3 HOMI Architecture

HOMI는 앞 절에서 지적하였던 이기종 미들웨어 인터페이스의 차이로 인한 연동 과정에서 발생하는 의미론적 문제를 해결하기 위해 단일한 인터페이스 중심으로 다양한 인터페이스를 브릿지하여 통합하는 방식을 회피하고, 사용자에게 연동 시나리오를 직접 작성할 수 있는 쉽고 간단한 인터프리터 언어를 제공함으로써 연동의 유연성과 호환 가능한 인터페이스 선택의 정확성을 높이게 하였다. 최종 사용자의 사용 형태를 충실히 반영할 수 있게 HOMI는 연동 시나리오를 간편한 방식으로 쉽게 구성할 수 있는 HOMIL(HOMI Language)이라는 인터프리터 언어를 제공한다. HOMIL은 복잡한 연동 시나리오도 특정 사건을 중심으로 일어나는 일련의 사건들의 흐름으로 표현할 수 있는, 사건 주도 처리(event-driven processing)방식의 스크립트 언어이다. 다음 그림 4는 HOMIL의 의사코드이다.

```
execute AService.Method1
execute BService.Method1 when AService.Method2
execute CService.Method2, DService.Method3 when BService.Method2
```

그림 4 HOMIL로작성한 의사 코드

HOMI의 동작 방식은 HOMIL로 작성한 언어를 파싱하고 when 절을 분석하여 유효성을 판별한 후, Event Manager를 통해 이벤트를 등록한다. HOMI는 등록된 특정 이벤트 발생을 감시하고, 이벤트 발생 시 execute절의 서비스를 실행하게 되며, 수행된 서비스는 다음 사건을 위한 이벤트를 발생시키며 이러한 사건들의 연속으로 특정 시나리오를 수행하게 된다. 특정 시나리오의 한번 등록되어지면 Configurator에 의해 관리 되어지며, 다음부터 같은 사건은 사용자의 개입 없이 이루어지게 된다. 이러한 처리 방식은 기존의 서비스별 브릿지 방식에서 오는 단점들을 제거하고 다양한 시나리오를 쉽게 구성할 수 있는 장점이 있다. 또한 HOMI는 이기종 미들웨어 간의 서비스 발견과 호출의 문제를 해결하기 위해 이기종 미들웨어의 현재 서비스되고 있는 혹은 새로 네트워크 내에 들어온 서비스를 보관하는 Repository를 두고 있다. Repository에 보관되는 형태는 CD(Common Descriptor)라는 XML형태의 HOMI 고유의 포맷이며, 각 미들웨어 Agent들은 자신의(UPnP, Jini, ...) 네트워크 내 장치들의 서비스를 발견하여 CD 형태로 변환하고, HOMI에게 전송하는 역할을 담당하며, 역으로 사용자가 HOMIL을 통해 만든 시나리오를 서비스 하기 위하여 CD포맷을 각 장치의 프로토콜 문법으로 변환하여 다른 미들웨어의 메소드를 호출하는 역할을 담당한다. 새로운 미들웨어가 출현하여도 CD포맷으로 변환하는 하나의 Agent를 만들어 이기종간 상호 호환성을 제공할 수 있고, 마찬가지로 HOMIL에 의해 제어될 수 있다. 다음은 HOMI의 주요 구성 요소들에 관한 설명이다.

(1) Middleware Agent : 장치들을 발견하고 HOMI에게 전송하며, HOMI에서의 명령을 각 장치들에게 전달하는 역할을 한다.

(2) Agent Manger : 각 미들웨어들의 Agent들을 관리하기 위한 일관된 인터페이스를 제공하며, Cache Manager와 Service Manager로 구성되어진다.

- Cache Manager

자주 사용되는 서비스들의 정보들을 캐시하고 각 미들웨어들의 주기적인 피드백을 통하여 사용자의 요청에 보다 빠른 응답을 하게 된다.

- Service Manager

한 장치의 동시 접근을 위한 동기화를 수행하며 등록된 서비스들을 관리한다. HOMI는 다양한 서비스들을 다음과 같은 상태로 관리한다.

- Register : 처음 서비스를 발견하고 등록되거나 Fail된 서비스가 등록되는 과정
- Ready : 서비스가 HOMI Service Manager에 등록을 마치고 정상적으로 동작하며 서비스 사용을 대기하고 있는 상태
- Use : 현재 서비스가 HOMI를 통해 사용중인 상태
- Fail : 해당 서비스가 무응답인 상태

(3) Web Service Provider : HOMI의 원격제어를 위한 API(Application Programming Interface)를 제공하는 계층으로서 가정 내에 존재하는 장치들을 원격에서 제어하고 반대로 밖의 웹 서비스를 로컬에 존재하는 장치들이 사용할 수 있게 변환기능을 제공한다.

(4) Configurator : 사용자가 등록하여 넣은 시나리오들을 관리한다.

(5) HOMIL Parser : HOMIL 스크립트 언어를 구문 분석하고 Event Manager에 Event를 등록하여 주는 일을 담당한다.

(6) Event Manger : HOMIL을 통해 들어온 이벤트를 구독(Subscribe)하고 이벤트를 관리하여 핸들러를 실행시켜준다.

(7) Communication Manager : 여러 가지 방식(IEEE1394, IrDA, LAN, PLC, Bluetooth)의 물리적 전송 방식을 통합하여 상위 계층에 단일 인터페이스를 제공한다

4. 결론 및 향후 과제

현재 홈 네트워크의 발전 형태를 고려하면, 통합 미들웨어는 반드시 필요하다. 하지만 홈 네트워크의 동적 특성을 최대한 반영하여 유연하게 구성되어 질 수 있는 구조를 가져야 하며, 사용자에게 다양한 시나리오를 편리하게 제공할 수 있는 방안이 필요하다. 본 논문에서는 그와 같은 문제들을 해결하기 위한 사용자 중심의 시나리오기반 통합 구조를 제안했다.

향후 과제는 현재 개발하고 있는 HOMIL의 편리한 작성을 위한 도움을 주는 시나리오 작성기 개발과 제어 응답 향상을 위한 캐시 매니저의 효율적인 알고리즘 등의 연구가 필요하다.

참고문헌

[1] UPnP Forum. <http://www.upnp.org>.
 [2] Sun Microsystems, " Jini Architecture Specification" . <http://www.sun.com/jini/>.
 [3] The Havi Organization " Havi Version 1.1 Specification" . <http://www.havi.org>.
 [4] Echelon Co., " LonTalk Protocol Specification ver 3.0," 1994.
 [5] Bill Rose, " Home Networks: A Standards Perspective," IEEE Communications Magazine, December 2001
 [6] Gerardo O' Driscoll, " The Essential guide to Home Networking Technologies," Prentice Hall, 2001
 [7] Henry Bequet, " Beginning Java Web Service" , WroxPress, 2003
 [8] Don Box, " Simple Object Access Protocol 1.1" available at URL " <http://www.w3.org/TR/SOAP/>"