

홈 네트워크 기기들의 상호 연동을 지원하는 규칙 기반 지능형 시스템 구조

A Rule Based Intelligent System Architecture for Interoperability Among Home Network Devices

이승륜*, 강순주**, 최준용**

경북대학교 정보통신학과*

경북대학교 전자공학과**

Seungryun Lee*, Soonju Kang**, Junyong Choi**

Department of Information and Communication, Kyungpook National University*

School of Electrical Engineering and Computer Science, Kyungpook National University**

E-mail : kissmark@ee.knu.ac.kr

ABSTRACT

홈 네트워크는 다양한 특성과 기능을 지닌 기기들로 구성된 전형적인 유비쿼터스 시스템이다. 그러나, 현재 홈 네트워크 환경에서 각 기기들은 대부분 독립적인 서비스를 제공하며, 기기들의 상호 연동을 지원하는 지능적인 서비스는 미비하다.

이와 관련하여 본 논문에서는 사용자가 손쉽게 홈 네트워크 상의 다양한 기기들의 상태를 파악하고 상호 연동이 가능한 제어 기능을 제공하는 지능형 시스템을 제안하고자 한다. 본 논문에서 제안하는 시스템은 크게 각 기기들의 고유 정보, 응용 분야에 적합한 서비스를 제공하는 규칙 기반 엔진, 사용자 인터페이스 부분으로 구성되어 있다. 다양한 기기들의 복잡한 제어를 규칙 기반 엔진이 담당함으로써 사용자는 단순한 명령만으로 정의된 규칙을 이용하여 홈 네트워크 기기들을 효율적으로 제어할 수 있다.

Key words : 홈 네트워크, 규칙 기반 엔진, 지능형 시스템

1. 서 론

홈 네트워크는 그 목적이나 제공하고자 하는 서비스에 따라, 다양한 특성과 기능을 지니고 있는 디바이스들로 구성된 전형적인 유비쿼터스 시스템이다.[5] 이러한 홈 네트워크 시스템은 제공하는 서비스에 따라, 멀티미디어 서비스, 데이터 통신 서비스, 가전 기기 제어 서비스 등으로 나뉘어 질 수 있다. 이러한 각각의 서비스 마다 상이한 디바이스로 구성되어 있으며 각각의 디바이스는 서비스를 구현하기 위한 개체 형태를 띄고 있다. 또한 그 특징은 디바이스의 복잡도와 역할, 목적에 따라 다르게 나타난다.

즉, 전구와 같이 조명을 조절하는 단순한 디바이스에서 HDTV, PVR과 같이 복잡한 멀티미디어 서비스를 제공하는 디바이스까지 수 많은 종류의 디바이스가 존재하며, 이들의 제어 방법이나 연결 관계는 서로 다른 특징을 지니고 있다. 이렇게 서로 다른 특성을 지니는 디바이스들을 제어하기 위해서 HAVi, Jini, UPnP[2]등 홈 네트워크용 미들웨어[1]가 개발되어 있지만, 각 서비스 별로 특화되어 있어서, 다양한 서비스를 제공하거나 서비스간 상호 작용을 하고 있지는 못하다. 그래서 현재 홈 네트워크 환경에서 동작하는 대부분의 디바이스들은 사용자의 지시에 따라 각각 독립적으로 제어되고 있으며, 그로 인해 디바이스간,

서비스간의 상호 연동을 통한 지능적인 서비스는 이루어지지 않고 있다. 따라서 홈 네트워크를 구성하고 있는 수많은 디바이스들 간의 여러 기능의 조합으로 이루어지는 복잡한 서비스를 제공하지 못하고 있으며, 사용자가 필요한 서비스와 기기의 조작 과정을 전부 이해하고, 순서대로 조작해야 하는 불편함을 지니고 있다.

이러한 사용자의 불편함을 해소 하기 위해서는 수많은 홈 네트워크 디바이스와 서비스를 포괄적으로 정의하고 제공하는 시스템이 필요하다. 제안하는 시스템 구조에서는 디바이스와 서비스를 분류하여 디바이스간 또는 서비스간 상호 연동이 유리한 구조를 제공한다. 또, 많은 수의 디바이스를 사용자 요구에 맞게 쉽고 편하게 연동시켜 작동시키기 위해서는 여러 가지 복잡한 케이스를 모두 만족시켜야 한다. 이러한 복잡한 규칙을 효율적이고 지능적으로 처리하기 위해서 규칙 기반 엔진을 정의하여 효율적이고, 확장성 있는 구조가 되도록 설계하였다.

II. 디바이스간 상호 연동을 위한 홈 네트워크 시스템의 고려 사항

홈 네트워크에서 디바이스간 상호 연동을 위해서는 우선 다른 환경에서 작동하는 디바이스를 동일한 환경에서 동작시키기 위해 공통적인 방법으로 표현 하는 일이 필요하다. 또 이렇게 표현된 디바이스에 각각 서로 다른 서비스를 제공할 수 있도록, 서비스를 분류하고 각 디바이스에 이 서비스를 등록시켜서 디바이스간 특성을 정의 할 필요가 있다. 마지막으로 이렇게 서비스에 필요한 특성이 주어진 디바이스를 효율적으로 연동하고 관리하기 위한 규칙 기반 엔진을 설계하여 디바이스간의 서비스가 서로 연동할 수 있도록 구현하여야 한다.

2.1 기존 연구

이렇게 홈 네트워크 환경에서 규칙 기반 엔진을 적용시킨 기존의 연구는 주로 사용자의 행동을 추론 하여 디바이스를 단순히 작동시키는 연구[4]였었다. 하지만, 제안하는 시스템에서는 규칙 기반 엔진이 규칙을 이용하여 디바이스간 연관성과 연동 가능성을 찾아 지능적으로 연동시키는 차이가 있다.

2.2 디바이스가 제공하는 서비스의 분류

상호 다른 서비스를 제공하는 디바이스가 산재하는 홈 네트워크 환경에서는 다양한 서비스와 프로토콜이 존재 하기 때문에 공통된 디바이스의 표현 방법이 필요하다.

디바이스를 표현하는 방법에는 여러 가지 방법이 있을 수 있는데, 홈 네트워크에서 디바이스간 연동을 위해서는 연관이 있는 디바이스 혹은 연관이 있는 서비스를 묶어서 관리 할 필요가 있다. 예를 들어 실내 조명을 제어 하는 커튼과 여러 종류의 전등은 같은 범주의 서비스로 관리 하여, 집안의 조명 관리를 손쉽게 할 수 있다. 즉, 홈 네트워크 내부에서 각 디바이스의 서비스를 표현하는 방법이 필요하다. 각각의 디바이스에서 제공하는 서비스를 미리 정의 하여 시스템 내부에서 발생하는 서비스를 파악하고, 파악된 서비스는 서비스간 연동과 서비스에 필요한 장비를 추가 삭제하는데 이용 되어야 한다.

2.3 디바이스 정의 및 분류

각 디바이스는 서로 다른 프로토콜과 기능을 지니고 있기 때문에, 공통된 시스템에서 작동하기 위해서는 공통적인 표현 방법이 있어야 하며, 디바이스간 연동을 위해서 디바이스가 제공하는 서비스 등의 필요한 정보를 스스로 가지고 있어야 한다. 기존 UPnP[2]등의 미들웨어는 XML을 기반으로 디바이스와 디바이스가 지원하는 서비스를 공통된 포맷으로 표현할 수 있으나, 명확하게 하드웨어를 제어 하는 방법에 대해 정의 하고 있지 않고, UPnP를 따르지 않거나 TCP/IP기반이 아닌 프로토콜에서 작동하는 디바이스는 표현하기 힘든 단점을 가지고 있다. 그러므로, 프로토콜에 무관하게 디바이스와 서비스를 표현할 수 있는 구조가 요구된다.

2.4 규칙 기반 엔진

다양한 디바이스와 서비스를 이용해서 사용자가 수행할 수 있는 연결은 디바이스의 수가 증가함에 따라 기하 급수적으로 증가 하게 된다. 이러한 연결을 효율적으로 관리하기 위해서는 연결 관계를 정리하여 간결하고 명확하게 표현하고, 쉽게 연결 관계를 추가하거나 삭제 할 수 있는 구조가 필요하다. 기존의 프로그래밍 언어로 이를 정의할 경우 디바이스의 수에 비해 기하 급수적으로 코드가 늘어 나게 되고, 연결 관계를 쉽게 표현하거나, 새로운 서비스나 디바이스에 대한 관계를 확장하는 것이 용의하지 못하다. 이러한 문제는 규칙 기반의 엔진을 이용하여 디바이스간 연결 관계와 의존 관계에 대한 규칙을 정의 함으로써 손쉽게 표현할 수 있다. 즉, 디바이스간 연결에 필요한 규칙을 정해 두고, 상황에 따라 적절한 연결을 제공해 주는 구조를 제공함으로써, 손쉽게 디바이스간 연동을 지원해 줄 수 있는 구조를 만

들 수 있다. 또한 새로운 디바이스나 서비스를 추가할 경우에도, 시스템의 전체 변화 없이 단지 규칙을 추가 및 수정하는 작업으로 시스템의 확장을 이룰 수 있다.

이렇게 전해진 명령은 다시 디바이스에 알맞은 형태로 변환되어 디바이스로 전해지고, 필요에 따라 유저 인터페이스를 통해 사용자에게 결과를 통보한다.

III. 제안 시스템

3.1 규칙 기반 홈 네트워크 시스템의 구조

그림 1은 본 논문에서 제안하는 규칙 기반 홈 네트워크 시스템 구조이다.

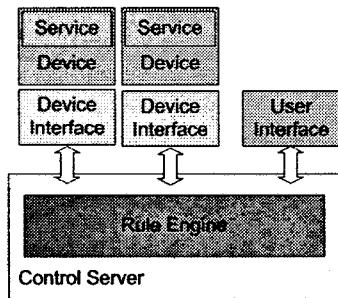


그림 1. 규칙 기반 홈 네트워크 시스템 구조

제안하는 홈 네트워크 시스템은 크게 전체 시스템의 동작을 제어하는 서버 부와 디바이스 부로 크게 나눌 수 있다.

디바이스 부는 실제의 디바이스 하드웨어를 추상화하고 디바이스 개체를 표현 하는 부분으로 모든 디바이스는 디바이스 인터페이스를 통하여 규칙 기반 엔진과 연결되게 된다.

서버 부는 사용자로부터 입력을 받아들이고, 서비스 수행결과를 다시 사용자에게 알려주는 사용자 인터페이스(User Interface)부분과 규칙 기반 엔진으로 나뉘어 진다. 규칙 기반 엔진에서는 사용자 인터페이스로부터의 입력과 디바이스의 상태 정보를 수집하여 규칙에 따라 적절한 서비스를 찾아 수행해준다.

3.2 규칙 기반 엔진의 설계

그림 2는 규칙 기반 엔진의 상세 구조이다. 모든 디바이스로부터의 정보 수집은 디바이스 인터페이스를 통해서 규칙 기반 엔진에서 처리하기 적합한 형태로 Working Memory에 저장된다. 이와 유사하게 사용자 인터페이스를 통해 사용자의 명령 또한 Working Memory로 저장된다.

디바이스의 제어와 상호 연동 과정에 필요한 규칙은 엔진의 외부에서 Rule Base로 공급되어 Working Memory와 Pattern Matcher에 의해 Rule Base와 Working Memory의 매칭이 일어난다. 매칭된 결과는 Agenda에 보관되었다가, Execution Engine에 의해 수행되어 다시 디바이스 인터페이스로 명령이 전해지며,

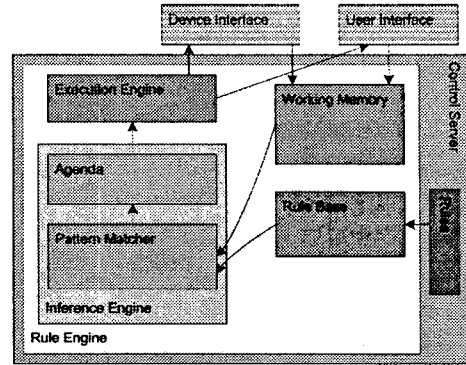


그림 2. 규칙 기반 엔진 구조

다음은 Rule Base의 설계 예로 커튼을 닫는 규칙의 일부이다. 커튼이 제공하는 서비스인 실내 조명 서비스와 관련이 있는 요소인 시간과 센서 값 그리고 다른 실내 조명 서비스 디바이스의 상태로부터 커튼이 취해야 할 행동을 취하게 되는 규칙이다.

```
(defrule curtain-close " curtain close "
(myClock (clock ?t&:(and (> ?t 6) (< ?t 18))))
(flamp (power1 0))
(sensorIn (light ?light&:(>= ?light 180) ))
(curtain (size ?s&:(> ?s 0)))
=>
(printout t crlf "Time: " (call ?myClock getClock) )
(printout t " -> curtain close:" ?s crlf)
(call ?curtain setSize -1))
```

3.3 디바이스와 규칙 기반 엔진의 작동 순서

그림3은 디바이스와 규칙 기반 엔진간의 데이터의 흐름을 보여주고 있다.

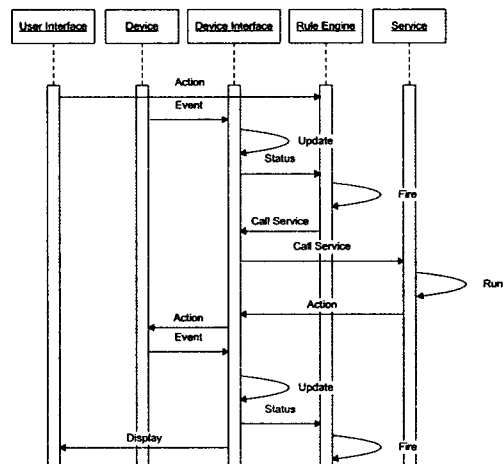


그림 3. 디바이스와 규칙 기반 엔진의 시퀀스 다이어그램

그림 3에서 보는 것과 같이 사용자 인터페이스로부터 입력이나, 디바이스 자체에서의 상태 변화에 따른 이벤트에 의해 디바이스의 변화된 상태는 디바이스 인터페이스에 의해 규칙 기반 엔진에 적합한 형태로 상태가 바뀌어서 규칙 기반 엔진으로 전달된다. 규칙 기반 엔진에서는 가능한 서비스를 매칭하여 조건에 따라 규칙을 수행하게 된다. 규칙에 따라 수행된 서비스는 디바이스 인터페이스를 통하여 디바이스에게 적절한 제어를 수행하게 되는데, 이렇게 제어에 의해 바뀌어진 상태는 사용자 인터페이스로 표시 하고, 다시 규칙 기반 엔진으로 상태가 전해져서 연속적인 규칙의 수행이 가능하도록 하여 디바이스간 연동 작업이 진행된다.

3.4 구현 및 결과

제안하는 홈 네트워크 시스템 구조는 Java Beans를 이용하여 디바이스 인터페이스를 정의하였으며, 규칙 기반 엔진에는 Java기반 규칙 기반 엔진 셀인 JESS[3]를 이용하여 규칙을 정의 하고 사용자 입출력을 처리하도록 하였다.

실제 입력을 받기 곤란한 센서와 주변 환경의 변화에 대한 데이터는 시뮬레이터를 작성하여, 가상의 환경에서 동작을 확인할 수 있도록 구현하였다.

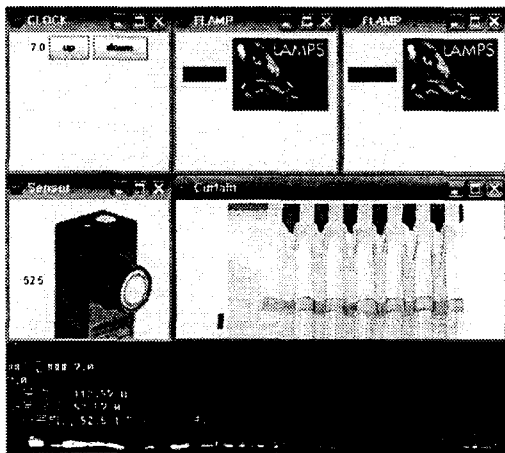


그림 4. 제안 시스템의 구현 예

그림 4는 규칙 기반 시스템을 구현한 것으로 램프와 커튼 그리고 광 센서를 연동하여 가정내 빛의 밝기를 조절하는 경우의 예이다. 그림 4에서 보듯이, 시뮬레이터를 통해 들어온 데이터와 사용자의 명령을 규칙을 이용하여 조합하여 수행하고 여러 디바이스를 연동시킬

수 있으며, 작동된 디바이스의 상태의 변화로 인해 관련 있는 디바이스가 규칙에 의해 지능적으로 연동하여 동작 한다.

IV. 결 론

본 논문에서는 사용자가 손쉽게 홈 네트워크 상의 다양한 디바이스의 상태를 파악하고 상호 연동이 가능한 제어 기능을 제공하는 지능형 시스템을 제공하기 위해서 규칙을 기반으로 하여 홈 네트워크 시스템을 구현하였다. 그 결과 다양한 디바이스들의 복잡한 제어를 규칙 기반 엔진이 담당함으로써 사용자는 단순한 명령만으로도 규칙을 이용하여 홈 네트워크 기기들을 제어할 수 있었으며, 디바이스의 추가에 따르는 복잡한 연관 관계를 규칙 기반 엔진의 규칙의 추가로 손쉽게 해결할 수 있었다.

Acknowledgements

본 연구는 한국 과학 재단 특정 기초 연구 (R01-2003-000-10252-0)의 연구비 지원으로 수행되었음.

V. 참고문헌

- [1] Soon Ju Kang, Jun Ho Park, Sung Ho Park, "ROOM-BRIDGE: A Vertically Configurable Network Architecture and Real-Time Middleware for Interoperability between Ubiquitous Consumer Devices in Home," Lecture Notes in Computer Science 2218, Page(s):232-251
- [2] Understanding Universal Plug and Play: A White Paper, UPnP Forum, Jun. 2000.
- [3] Ernest Friedman-Hill, "Jess In Action", Manning Publications Co.,2003.
- [4] 최종화, "지능적인 홈을 위한 상황인식 미들웨어에 대한 연구", 한국정보처리학회 논문지A, 2004. December, Volume: 11-A, No.7 Page(s):529-536
- [5] Stephen S. Intille, "Designing a Home of the Future", IEEE Pervasive Computing,2002. April, Volume: 1, Issue 2, Page(s):76-82