
AHAM CHA 기반 객체지향 협업 모델

김용호* · 이종환** · 유동희***

Object Oriented Cooperation Model based on AHAM CHA

Yong-ho Kim* · Jong-hwan Lee** · Dong-hui Yu***

요약

스마트 홈이나 지능화된 홈 네트워크 제어 시스템이 거주자에 대한 자동화된 서비스를 제공하기 위해서 홈 디바이스들의 협업을 필요로 한다. 홈 디바이스들의 협업은 사용자의 제어 요구에 대해 같은 목적을 가지는 둘 이상의 홈 디바이스들을 동시에 제어하는 거주자를 위한 서비스로 정의할 수 있다. 홈 디바이스 협업의 대상이 되는 것은 홈 디바이스의 동작 객체이며, 이 객체들은 AHAM(Association of Home Appliance Manufactures) CHA(Connected Home Appliances)에 디바이스(가전기기) 모델 별로 정의되어 있다. AHAM의 CHA는 일반적인 가전기기 모델, 객체들과 상위 수준의 메시지들을 표현함으로써 네트워킹을 통해 가능한 서비스와 특징들을 활성화하고자 하고 있으나, 디바이스 모델의 객체간의 협업의 표현과 처리에 대해서는 고려되어 있지 않다. 본 논문은 홈 디바이스들 사이에서 발견할 수 있는 협업 관계를 정의하고, AHAM의 객체들이 협업 관계를 처리할 수 있는 협업 모델을 제안한다.

ABSTRACT

The Smart Home or intelligent home network control system has the need for home devices to cooperate for the purpose of providing the automated services to residents. The cooperation of home devices can be defined as the service for residents that it controls simultaneously two or more home devices with the same purpose about user's control request of specific home device. A object of home device cooperation are the function object of home device, it is well defined in each device model in AHAM(Association of Home Appliance Manufactures) CHA(Connected Home Appliances). AHAM CHA is to promote new appliance services and features enabled through networking by describing generic appliance models, objects, and high-level messages, but does not consider the description and process of cooperation between objects of each device model. This paper defines the cooperation relationship that could be discovered in among home devices, and proposes the cooperation model that AHAM objects could process the cooperation relationship.

키워드

AHAM, 디바이스 모델, 협업 관계, 객체지향 협업 모델

1. 서론

스마트 홈이나 지능형 홈 네트워크 제어 시스템의 사용자는 소비자들이며, 소비자들은 스마트 홈 시스템의 도입을 통해 이전에 비해 더 편리하

고 안락한 생활을 하기를 원한다. 스마트 홈을 사용하는 소비자들은 자신이 선호하는 회사의 제품을 구입하여 가정 내에 구축된 홈 네트워크에 쉽게 연결되기를 원하고, 네트워크에 연결된 홈 디바이스들이 서로 통신하는데 문제가 발생하기를

* 한국기계연구원 자본재정보기술그룹 선임연구원

** 부산대학교 컴퓨터및정보통신연구소 전임연구원

접수일자 : 2003. 9. 24

*** 부산가톨릭대학교 컴퓨터정보공학부 조교수

원하지 않으며, 에어컨을 켜면 창문과 문을 닫는 다든지, 또는 홈 씨어터 시스템을 통해 영화를 시청할 때 조명 시스템이 영화 모드로 바뀌고 에어컨이 동작하여 영화 시청에 쾌적한 분위기를 자동적으로 설정하는 형태의 서비스를 스마트 홈 시스템이 제공하기를 원하는 것이다[1,2].

스마트 홈이 홈 디바이스 제어가 가능하기 위해 필요한 홈 네트워크 형 가전기기들이 상용화되고 홈 네트워크 플랫폼과 관련 제어 프로토콜 표준이 안정화되고 있는 추세이며, 각 기업들은 표준화된 제어 프로토콜을 이용하여 자사의 제품에 대해 초기 수준의 스마트 홈 기술을 개발하여 각 가정에 공급하고 있다[3,4,5,6,7]. 홈 네트워크 제어 통신을 위한 표준은 상당한 성과를 이루었으나 제어 대상이 되는 각 홈 디바이스들이 표준화되어 있지 않기 때문에 각 가정에서 각기 다른 기업의 가전기기를 구입하였을 때 홈 네트워크와의 연결뿐만 아니라 원격 제어 등 원활하게 연동되지 않을 가능성이 높다. 이에 스마트 홈 사업을 추진하고자 하는 가전 기업들은 홈 디바이스 표준 모델의 필요성을 공감하고 AHAM(Association of Home Appliance Manufactures)이라는 협의체를 만들어 홈 디바이스 표준 모델 제정을 추진 중이다. AHAM은 세계 우수 기업들이 참여하여 만든 것으로서, CHA(Connected Home Appliances)에 대해 메시징 관점에서 표준화를 위해 2000년 8월부터 SATF(Smart Appliance Task Force)를 만들어 활동하고 있다. 초기에는 가전기기과 통신 네트워크 사이의 인터페이스에서 스마트 가전기기의 상호 운용성을 확보하기 위해 의도적인 합의를 도출한 표준을 만드는 데 주력하였고 그 결과로 AHAM CHA-1-2003를 작성하여 7개의 디바이스(가전기기) 모델에 대한 표준안을 만들어 배포하고 있다. 이것은 ANSI 표준으로 채택되었고 홈 네트워크 제어 규격인 UPnP나 OSGi 등에서도 활용하기 위해 적극적 검토를 추진 중이며, EIA나 ISO 등으로 확대될 전망이다[8,9].

스마트 홈이나 지능화된 홈 네트워크 제어 시스템이 거주자에 대한 자동화된 서비스를 제공하기 위해서 홈 디바이스들을 서로 협업하게 한다 [1,2,10,11]. 홈 디바이스들의 협업이란 거주자에게

서비스를 제공하는 같은 목적을 가지는 둘 이상의 홈 디바이스들이 동시에 동작하는 것으로 정의할 수 있다. 예를 들면, 거주자가 퇴근하여 집의 현관에 들어서면, 현관의 조명이 켜지고 목욕물을 데우고 음악이 켜지는 등의 서비스를 스마트 홈이 자동적으로 제공할 수 있다. 조명기기, 보일러와 오디오 장치들이 서로 협력하여 거주자에게 서비스하도록 협업 제어를 받는 것이다. 이와 같이 홈 디바이스의 협업을 통해 거주자에 대한 서비스 효율이 높아진다.

협업의 대상이 되는 것은 홈 디바이스와 동작 객체이다. 이 홈 디바이스의 동작 객체의 명칭과 내용을 참조하여 홈 디바이스가 어떤 동작을 하는 것인지 의미를 파악할 수 있어야 한다. 이를 위해 홈 디바이스의 동작 객체에 대해 대 다수의 기업이 인정하는 표준을 따르는 것이 유리하다. 이 동작 객체에 대해서는 AHAM의 디바이스 모델에 잘 정의되어 있다. 그러나 AHAM은 하나의 디바이스 내에 정의할 수 있는 데이터와 기능을 처리하고 수행하는 객체에 대해서는 잘 정의되어 있지만, 객체들 간 협업에 대해서는 고려되어 있지 않다.

본 논문에서는 홈 디바이스 모델의 동작 객체 간 협업 관계를 정리하고 AHAM 디바이스 모델의 객체가 협업 관계에 있는 다른 디바이스 모델의 객체와의 협업 관계를 설정하고 참조할 수 있는 AHAM기반의 협업 모델을 제안한다.

II. AHAM CHA - 객체 모델링

AHAM CHA는 가전기기 또는 홈 디바이스들이 홈 시스템 네트워크에 연결되어지는 통신 인터페이스 모듈을 가지고 있다는 가정 하에, 일반 가정에서 사용하는 가전기기에 대한 모델들, 객체들과 상위 수준의 메시지들을 기술함으로써 네트워크를 통해 서비스할 수 있는 새로운 형태의 가전기기 서비스와 특징들을 창출하기 위한 목적으로 만들어진 것이다. AHAM의 디바이스 모델들은 사용자, 서비스 제공자, 그리고 다른 디바이스들이 원격으로 접근하고 제어하는데 활용할 수 있도

록 각 가전기기와 가전기기의 동작들을 객체화하여 표준화된 모델을 제안하고 있다[8].

AHAM은 디바이스들을 모델링하기 위해 객체지향 설계 방법을 사용하고, 각각의 AHAM 디바이스들은 디바이스를 구성하는 데이터와 기능 객체들을 정의하고 분류함으로써 하나의 디바이스 모델로 정의된다. AHAM 디바이스 모델의 객체들은 몇 개의 부시스템(Subsystem)으로 편성되는데, 이 객체들은 AHAM에서 제공하는 10개의 객체 클래스들로부터 만들어진다. 이 객체 클래스가 지원하는 메서드들은 GetValue, SetValue, SetOn, 그리고 SetOff 등이 있다.

2003년까지 AHAM은 Clothes Washer Appliance Model, Refrigerator-Freezer Appliance Model, Clothes Dryer Appliance Model, Dishwasher Appliance, Range, Oven, Cooktop Appliance Model, Countertop Microwave Oven Appliance Model, 그리고 Room Air Conditioner(Window Type) Appliance Model의 7가지 홈 디바이스만 정의하고 있으며, 일반적인 가전기기 모델, 객체와 상위 수준의 메시지와 확장성을 고려한 템플릿을 제공하고 있다. 스마트 홈에 적용되기 위해서는 정의할 수 모든 디바이스에 대해 디바이스 모델을 정의하여야 하는데, 향후에는 일반 가전기기뿐만 아니라 창문, 조명기기, 방문 등 집을 구성하는 모든 디바이스를 대상으로 표준화해 나갈 것이다.

AHAM에서 제시하는 디바이스 모델의 표현 구조와 계층 구조는 그림 1과 그림 2와 같다. 하나의 디바이스는 시스템, 서브시스템, 객체 컨테이너, 객체와 프라퍼티로 구성된 구조로 표현된다. 그림 1, 그림 2에서 디바이스는 에어컨, 전자레인지 등 가전기기를 의미하고, 시스템은 디바이스에서 주요 기능을 말하는 것으로써, 하나 이상의 주요 기능을 제공하는 디바이스는 시스템 레벨을 가진다. 하나 이상의 시스템을 갖는 경우의 예를 들면, “cooktop” 시스템과 “oven” 시스템으로 “range” 디바이스가 있다. 디바이스의 주요 기능이 하나인 경우는 시스템 레벨을 정의하지 않으며, 이름을 구성할 때 무시한다. 부 시스템은 디바이스의 시스템을 구성하는 레벨이다. AHAM에서

사용되는 서브시스템의 예로는, “User Control” 부 시스템, “Operation Monitoring” 부 시스템 등이 있다. 객체 컨테이너는 논리적으로 관련이 있는 객체들을 모으기 위해 사용되는 것이다.

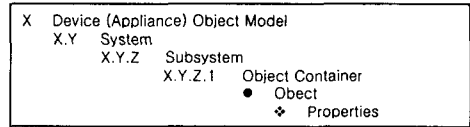


그림 1. 디바이스(가전기기) 모델들의 표현구조
Fig. 1 Structure of Device(Appliance) Models

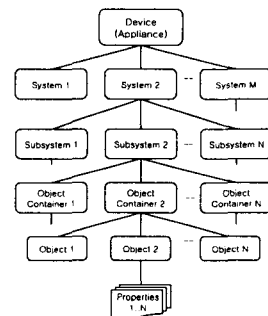


그림 2. AHAM 디바이스(가전기기) 모델 계층구조
Fig. 2 AHAM Device(Appliance) Model Hierarchy

AHAM CHA 문서는 제한된 디바이스 모델과 동작 객체만을 모델링하고 있어, 협업을 적용하기 위해서는 협업과 관련한 추가적인 객체가 필요하다. 본 논문에서는 추가적인 객체의 정의와 협업을 처리하는 방안을 제시한다.

III. 홈디바이스의 협업과 협업 관계

홈 디바이스의 협업은 둘 이상의 독립적인 홈 디바이스가 거주자에게 서비스를 제공하기 위해 함께 동작하는 것이다. 그러나 서비스의 목적이 다양하고 거주자의 특성이나 생활 패턴에 따라 다르기 때문에 일반화하기는 어렵다.

홈 디바이스들은 효율성과 편의성 측면에서 상호 의존적인 관계에 있다. 이러한 의존적인 관계를 본 논문에서는 홈 디바이스의 협력관계에 있다고 정의한다. 예를 들어, “에어컨을 켜면 열효율을 위해 창문과 문을 닫고 난방기구의 운영을 중단한

다.” 라는 협력관계가 있을 수 있다. 이 때 에어컨, 창문, 문과 난방기구는 협업을 하게 되고 디바이스들은 협력 관계에 있다고 말할 수 있다.

디바이스 간 의존성 관점에서 두 디바이스 관계는 의존적 협력 관계, 배타적 협력 관계와 독립적 관계의 세 가지로 나눌 수 있다. 이 세 가지 관계를 이용하여 두 디바이스 A와 B의 관계를 정리하면 다음의 세 가지 유형으로 나누어 볼 수 있다.

협력1관계) A → B : B는 A에 대해 의존적 협력 관계. A가 어떤 목적을 위해 활성화 되면, 같은 목적을 가지는 B도 활성화 되어야 한다.

협력2관계) A ← B : B는 A에 대한 배타적 협력 관계. A가 어떤 목적을 위해 활성화 되면, B는 반대의 목적을 가지기 때문에 비활성화 되어야 한다.

협력3관계) A, B : A와 B는 독립적 관계, 또는 아무 관계 아님.

서비스 측면에서 보면 디바이스는 거주자를 위해 임의의 동작을 하여야 한다. 따라서 앞서 말한 협력 관계만으로는 구체적인 서비스 내용을 나타낼 수 없다. 예를 들어, 에어컨과 창문은 협력1관계에 있다고 할 수 하지만 이 관계만으로는 거주자에게 제공하고자 하는 서비스에 대해 디바이스들이 어떻게 동작해야 하는지를 결정할 수 없다. 이를 해결하기 위해서는 두 디바이스 사이의 상호의존성을 표현할 때, 디바이스의 동작이나 상태까지 고려하여야 한다. 협력 관계에 있는 두 디바이스들의 동작(또는 상태)들에 의해 좀 더 구체적인 관계를 설정할 수 있다. 이 관계를 본 논문에서는 협업 관계라고 정의한다. 에어컨과 창문의 협력1관계에서 에어컨과 창문의 동작이나 상태를 고려한 협업 서비스를 예시하면, “에어컨이 켜져 있는 상태이면 창문을 닫는 것이 좋다.”의 형태로 기술할 수 있다. 이 예시는 거주자에게 직접적으로 제공하는 서비스의 내용을 담고 있다. 이것을 상호의존성 표현 방법을 이용하여 나타내면 다음과 같이 할 수 있다.

에어컨.켄 → 창문.닫음

마찬가지로 상태를 기준으로 의존성을 표현하면 아래

와 같다.

에어컨.냉방중 → 창문.닫혀있음

만약 에어컨을 켜는 동작과 창문을 여는 동작에 대해 의존성을 표현한다면,

에어컨.켄 ← 창문.열

이 될 것이다. 따라서 디바이스 사이의 의존적 관계를 구체적으로 표현하기 위해 디바이스의 동작이나 상태까지 고려하는 것이 의미상 타당하다. 디바이스 A와 B의 관계가 B는 A에 대해 의존적 관계이고, A가 {a1, a2, a3, a4, a5} 이라는 동작이나 상태가 있고, B가 {b1, b2, b3, b4}의 동작이나 상태가 있다고 하자. 이때 A.{a1, a2, a3}과 B.{b1, b3}의 쌍과 {a4}와 {b2}의 쌍이 각각 협력적인 협업관계, A.{a1, a2}와 B.{b2}는 비협력 협업관계, 그리고 A.{a5}와 B.{b4}는 독립적인 관계라고 말할 수 있다. 이것을 의존성 표현 방법을 사용하여 나타내면 다음과 같이 할 수 있다.

A → B
 A.{a1,a2,a3} → B.{b1,b3}
 A.{a4} → B.{b2}
 A.{a1,a2,a3} ← B.{b2}
 A.{a4} ← B.{b1, b3}
 A.{a5}
 B.{b4}

동작에 대한 제어의 측면에서 보면 비협력 협업 관계와 독립적인 관계는 관계로는 분류될 수 있지만, 실제 동작을 요구하는 내용은 아니다. 에어컨과 창문의 동작을 고려한 관계를 예로 들어 보면, ‘에어컨 → 창문’의 관계 하에서 에어컨에 대해서는 {전원ON, 냉방시작, 냉방중지, 전원OFF}라는 기본 동작이 있고, 창문은 {닫음, 열}의 기본 동작이 있다고 가정하자. 이때, 에어컨의 {냉방시작}과 창문의 {닫음}은 협력적 관계, 에어컨의 {냉방시작}과 창문의 {열}은 비협력 관계, 그리고 에어컨의 {전원ON, 냉방중지, 전원OFF}와 창문의 {열}은 독립적 관계이다. 이것을 의존성 표

현 방법을 사용하여 나타내면

```
에어컨 -> 창문
에어컨.냉방시작 -> 창문.닫음
에어컨.냉방시작 -< 창문.열
에어컨.(전원ON,냉방중지,전원OFF)
창문.열
```

으로 표현된다. 이 때 “에어컨->창문”의 관계에 따라 “에어컨.냉방시작->창문.닫음”만 실제 동작의 제어 처리에서 의미가 있으며 나머지는 관계로써만 의미를 가진다. 따라서 동작의 제어라는 측면에서는 의존적 협력 관계에 있는 디바이스의 동작들을 대상으로 협력적인 협업 관계에 있다고 할 수 있고, 이에 따른 협업 관계의 처리는 아래와 같이 정리할 수 있다.

협업관계) A.a -> B.b : B의 동작 b는 A의 동작 a에 대해 의존적인 관계이며 A.a와 B.b는 협업관계이다. A.a가 어떤 목적을 위해 활성화 되면, 같은 목적을 가지는 B.b가 활성화 된다.

협업관계의 처리 두 디바이스 A와 B가 협력 관계이고, A는 {a1} 동작이 있고, 디바이스 B는 {b1, b2} 동작이 있다. A.a1->B.b1, A.a1->B.b2 이면 A.a1에 대해 B.b1과 B.b2에 대한 정보를 저장하여 A.a1이 활성화될 때 B.b1, B.b2도 활성화 되도록 한다.

3장에서 정리한 협업관계는 홈 디바이스의 동작들 간의 협력 여부에 의해 설정된다. 홈 디바이스의 동작은 AHAM CHA에서 제시하는 디바이스 모델을 구성하는 객체들 중 사용자의 제어와 관련한 객체들과 일치한다. 4장에서는 AHAM에서 정의하는 객체를 3장에서 제시한 협업관계와 처리 방법을 반영한 객체 지향 협업 모델을 제안한다.

IV. 홈 디바이스 협업 모델

본 논문에서 제안하는 협업 모델은 AHAM 디바이스 모델을 구성하는 객체들 중 사용자 제어 대상이 되는 객체들이 협업관계를 처리할 수 있도록 하는 것이다. 협업 관계 처리를 위해 새로운 클래스를 정의하고, 새로 정의한 클래스와 AHAM

디바이스 모델에서 제어 동작을 정의하는 클래스를 이용하여 기존의 객체를 협업 객체로 만들어서, 협업 관계에 있는 객체의 정보를 참조하고 호출할 수 있도록 한다. 4.1절에서는 협업 관계를 처리하는 클래스 정의, 4.2절은 AHAM의 객체에게 협업 정보를 추가하는 방법을 보여주고, 4.3절에서는 협업 모델의 적용 내용을 설명한다.

4.1 협업 관계를 처리하는 클래스 정의

AHAM의 객체 클래스들은 각 디바이스(가전기기) 모델들의 독립적인 기능과 데이터를 포함하는 객체를 정의할 수가 있지만, 협업 관계 및 동작을 처리하는 부분에 대해서는 정의하지 않고 있다. 따라서 본 논문에서는 AHAM에서 협업 객체를 정의할 수 있도록 협업 클래스를 정의한다. 이 협업 클래스는 기존의 AHAM 객체 클래스를 상속받도록 하여 AHAM 객체의 특성을 그대로 유지하도록 한다.

협업 클래스는 협업 관계에 있는 객체의 참조 정보 관리, 참조 정보를 이용하여 협업 관계에 있는 객체의 메서드를 호출할 수 있도록 설계됨으로써 3장의 협업 관계를 처리할 수 있도록 구성된다.

AHAM에서 제공하는 객체 클래스들 중에 협업과 관련하여 사용될 수 있는 것은 Binary Control 객체 클래스이다. 이 클래스로 선언되는 객체는 ON과 OFF의 두 가지 중 하나의 상태로 결정되며 ON 상태 값에 의해 디바이스가 활성화된다. 상태 결정을 위해 이 클래스는 SetOn과 SetOff 메서드를 지원하며 인자 값은 TRUE 또는 FALSE를 사용한다.

본 논문에서 제안하는 협업 모델은 AHAM에서 정의하는 디바이스 모델의 객체가 협업 관계를 처리할 수 있도록 하기 위해 협업 클래스를 필요로 한다. 협업 클래스는 아래와 같이 정의된다.

```
class CooperativeAction : Public BinaryControl (
private
    CooperativeObjectInfo co_actions[];
public
    Append(CooperativeObjectInfo new_obj )
    {
```

```

        // new_obj를 co_actions에 추가한다
    }
    Delete(CooperativeObjectInfo obj )
    {
        // co_actions있는 obj를 삭제한다
    }
    InvokeAll()
    {
        // co_actions의 객체정보를 하나씩 읽어서
        // StartAction()을 호출한다.
    }
    StartAction(CooperativeObjectInfo obj)
    {
        // obj 객체의 SetOn 메시지를 호출한다.
    }
}

```

새로 정의하는 CooperativeAction 클래스가 상속하는 BinaryControl 객체 클래스는 AHAM에서 제공하는 사용자 제어 동작을 선언할 수 있는 객체 클래스이다. 협업 관계에 있는 객체의 참조 정보는 co_actions에 저장되어 처리된다. 이 협업 객체의 참조 정보는 아래의 CooperativeObjectInfo 클래스를 이용하여 선언한다.

```

class CooperativeObjectInfo {
public
    string      CoopObjectName;
}

```

CooperativeAction 클래스는 CooperativeObjectInfo 클래스의 CoopObjectName에 협업 관계에 있는 대상 객체 정보를 Append, Delete 메시지를 이용하여 객체 정보를 추가 및 삭제하고, StartAction, InvokeAll 메시지를 이용하여 협업 객체들을 호출하도록 구성되어 있다. CoopObjectName은 문자열 데이터를 저장된다. 문자열 데이터의 값은 AHAM의 디바이스 모델의 표현 구조에 의해 디바이스 모델 이름, 시스템 이름, 부 시스템 이름, 객체 컨테이너 이름과 객체 이름을 붙여서 하나의 문자열로 만들면 된다. 협업 관계에 있는 객체가 어느 디바이스의 어떤 객체인지 알 수 있는 정보를 저장하는 것이다. 예를 들어, “에어컨을 켜다”는 객체를 CoopObjectName에 넣는다면, “AirConditioner. UserControl.Environment-Control.StartCycle”의 값이 저장된다. 예를 든 에어컨의 경우 시스템 항목의 값이 생략되어 있는데, AHAM에서는 시스템 항목이 2개 이상으로

나누어지지 않는 경우 대상 객체의 이름을 조합할 때 무시한다. 4.2절에서는 기존의 AHAM의 객체에게 새로 정의한 클래스를 이용하여 협업 정보를 추가하는 방법에 대해 살펴본다.

4.2 AHAM의 객체에게 협업 정보를 추가하는 방법

CooperativeAction 클래스는 AHAM의 객체 클래스인 Binacry Control 클래스를 상속받아 협업 관계에 있는 객체들을 참조하도록 되어 있다. 보기를 들면, 에어컨을 동작시키는 Start Cycle 객체는 Binary Control 클래스로부터 만들어진다. 그러나 Binary Control 클래스는 협업 정보를 나타낼 수 없으므로, 앞 절에서 정의한 협업 관계에 있는 객체 참조 정보를 관리할 수 있는 CooperativeAction 클래스가 Binary Control 클래스를 상속받음으로써 협업 제어 가능한 클래스를 정의한다.

```

Class CooperativeAction inherits BinaryControl;

```

AHAM 의 에어컨 가전기기 모델에서 에어컨 동작을 시작하는 Start Cycle 객체는 CooperativeAction 클래스로 새로 정의하여 협업 관계가 처리 가능한 객체로 만든다. 이 과정은 다음과 같은 코드로 표현할 수 있다.

```

CooperativeAction StartCycle;

```

새로운 클래스로 만들어진 StartCycle 객체는 협업 정보를 처리할 수 있다. 3장에서 나타낸 협업 관계에서 “에어컨.컴->창문.염”을 추가하기 위해서는 StartCycle 객체의 Append 메시지를 호출함으로써 협업 정보를 추가할 수 있다.

```

StartCycle.Append(Window.WindowControl.WindowClose)

```

Append 메시드에 의해 저장되는 값들은 거주자에게 협업 정보 설정 인터페이스를 제공하여 처리하거나, 거주자의 홈 디바이스 사용 패턴을 분석하여 발견되는 협업 정보를 AHAM 디바이스 모델에 맞도록 변환하여 AHAM 기반 협업 정보

를 생성할 수 있다.

StartCycle 객체의 Delete 메서드를 호출하면 협업 정보를 삭제할 수 있다.

```
StartCycle.Delete(Window.WindowControl.WindowClose)
```

Append 메서드로 협업 정보를 추가했다면, StartCycle.InvokeAll()을 호출함으로써 협업 관계에 있는 객체 참조 정보를 하나씩 StartAction() 메서드를 이용하여 호출한다.

4.3 협업 시나리오에 대한 협업 모델의 적용

홈 디바이스의 협업은 거주자에 대해 편의성이나 효율성을 제공하기 위해 동시에 동작할 필요성이 있는 홈 디바이스들이 함께 동작을 하는 거주자를 위한 서비스이다. 홈 디바이스들 간의 협업에 대한 시나리오는 가전기기 판매자로부터 정보를 제공받거나, 거주자 생활 패턴 분석, 각 집안의 가전기기 사용 로그의 분석에 따른 패턴 분석을 통해 추출할 수 있다[1,2,12,13]. 본 논문에서는 여러 가지 유형의 협업 시나리오 중에서 거주자에 대한 공조 서비스와 관련한 협업 시나리오를 중심으로 협업 모델을 적용하고자 한다. 공조기 서비스에 대한 협업 시나리오를 간단히 하면 아래와 같다.

“에어컨을 켜면 열효율을 위해 창문과 문을 닫고 난방기구의 운영을 중단한다.”

이 시나리오는 에어컨, 창문, 문, 그리고 난방기구의 디바이스 모델을 필요로 한다. AHAM은 이 네 가지 중 에어컨에 대한 디바이스 모델만 정의되어 있다. AHAM의 디바이스 모델링 방법을 이용하면 AHAM에서 정의하지 않은 다른 디바이스의 모델을 정의할 수 있다. 본 절에서는 AHAM에서 이미 정의한 디바이스 모델인 AirConditioner와 Window, Door, 그리고 Heater 디바이스 모델을 객체 모델링 방법으로 정의하여 협업 모델을 적용한다. 홈 디바이스 동작들에 의한 협업 관계는 사용자에게 의한 디바이스 제어 동작과 연관지어

져 있다. 따라서 협업 모델에서는 AHAM의 디바이스 모델에서 사용자가 직접 제어 가능한 기능을 정의하는 부분만을 필요로 한다. 에어컨의 경우는 User Control Subsystem에서 Environment Control Object Container에 담겨져 있는 객체들이 이에 해당된다. 이 객체들 중 특히 Binary Control class에 의해 만들어진 객체들인 Start Cycle, Ready for Start, Pause Cycle, Cancel Cycle, 그리고 Fan Only Mode 객체들이 협업 모델을 적용할 수 있는 대상이 된다. 본 논문의 협업 모델이 필요로 하는 디바이스 모델인 Window, Door, 그리고 Heater에 대해서도 사용자 직접 제어 영역에 대한 부분만을 정의한다. 먼저 Window 디바이스 모델에 대한 User Control 부 시스템에서 사용자 제어 영역을 정의하면 그림 3과 같이 할 수 있다.

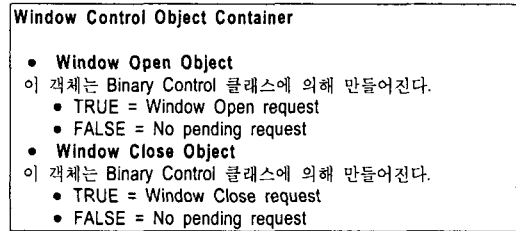


그림 3. Window 디바이스 모델의 사용자 제어 부 시스템의 일부
Fig. 3 Part of User Control Subsystem of Window Device Model

Door와 Heater 디바이스 모델에 대한 User Control 부 시스템의 사용자 제어 영역을 정의하면 각각 그림 4, 그림 5와 같이 선언할 수 있다.

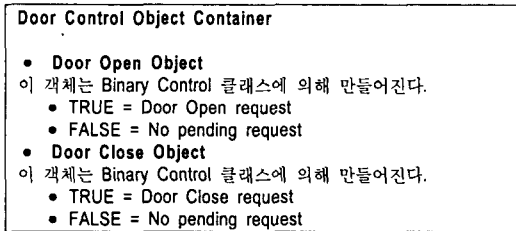


그림 4. 문(Door) 디바이스 모델의 사용자 제어 영역
Fig. 4 Part of User Control Subsystem of Door Device Model

여기까지는 AHAM에서 제시하는 제어 객체 클래스에 의해 객체들을 만든 것이다. 협업 객체를 정의하고자 한다면, 각 디바이스 모델의 객체 중 Binary Control 클래스에 의해 정의된 객체들을 CooperativeAction 클래스에 의해 다시 정의한다. 이렇게 함으로써 협업 관계에 있는 객체를 참조하여 협업 처리를 할 수 있다. Window 디바이스 모델의 객체를 CooperativeAction 클래스로 정의하면 나타내면 그림 6과 같다.

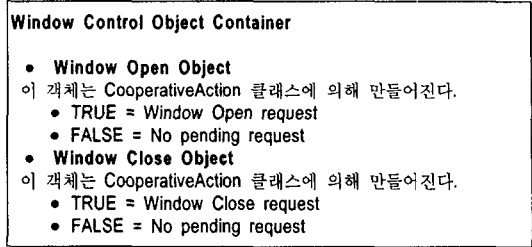


그림 6. 그림 3의 객체를 CooperativeAction 클래스에 의해 다시 정의함

Fig. 6 Object of Fig. 3 redeclared by CooperativeAction class

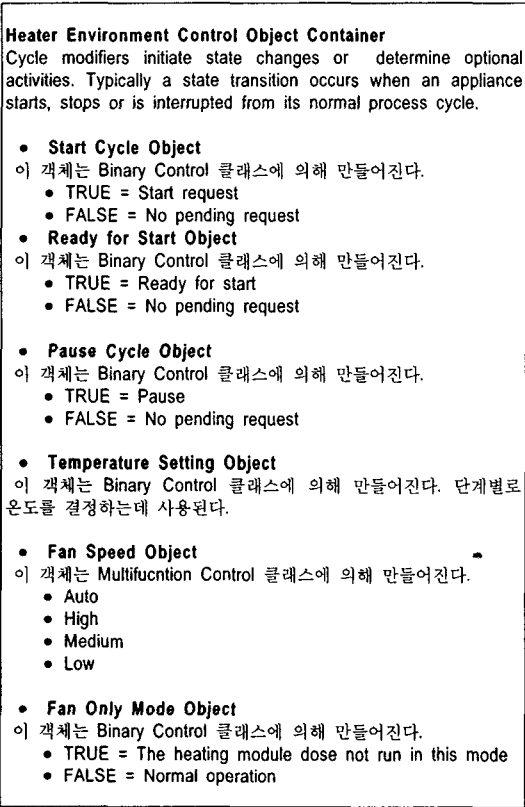


그림 5. Heater 디바이스 모델의 사용자 제어 영역
Fig. 5 Part of User Control Subsystem of Heater Device Model

협업 객체는 AHAM의 객체와 같은 이름으로 선언되며, 그 특성은 그대로 상속되기 때문에 기존의 AHAM의 객체와 차이가 없고, 협업관계 처리 기능만 추가된 것이다.

협업 객체를 정의하였으면 협업 관계에 있는 객체 참조 정보를 추가하여야 한다. 협업 시나리오에 의한 협업 정보를 3장의 협업 관계로 도식화하면 아래와 같다.

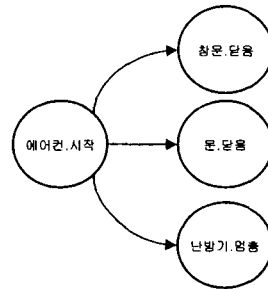


그림 7. 디바이스 동작 객체들 간 협업 관계
Fig. 7 Cooperation relations among objects of each device model

그림 7의 각 디바이스 동작을 나타내는 객체는 AHAM의 객체 모델링 방법에 따라 정한 객체의 표현방법으로 표 1에 나타낸 바와 같이 변환된다.

그림 7에 따라 “창문.닫음”, “문.닫음”과 “난방기.멈춤”에 관련한 객체들은 “에어컨.시작” 객체의 협업 정보로 저장되어야 한다. 즉, AirConditioner.UserControl.EnvironmentControl.StartCycle 객체의 협업 관계 객체 정보 저장 변수에 세 가지 객체의 참조 정보를 등록하게 되고 이것은 아래와 같다.

```
AirConditioner.UserControl.EnvironmentControl.StartCycle.A
ppend(Window.WindowControl.WindowClose)
```


AirConditioner.UserControl.EnvironmentControl.StartCycle.Append(Door.DoorControl.DoorClose)

AirConditioner.UserControl.EnvironmentControl.StartCycle.Append(Heater.UserControl.EnvironmentControl.PauseCycle)

만약 등록된 협업 객체 중, “난방기.멈춤” 객체를 삭제한다면 아래와 같이 할 수 있다.

AirConditioner.UserControl.EnvironmentControl.StartCycle.Delete(Heater.UserControl.EnvironmentControl.PauseCycle)

이터를 기준으로 StartAction 메서드를 호출한다. 호출한 StartAction 메서드를 통해 호출 되는 것은 각각 아래와 같은 내용이 된다.

Window.WindowControl.WindowClose.SetOn(TRUE)
 Door.DoorControl.DoorClose.SetOn(TRUE)
 Heater.UserControl.EnvironmentControl.PauseCycle.SetOn(TRUE)

이렇게 함으로써 “에어컨을 켜면 열효율을 위해 창문과 문을 닫고 난방기구의 운영을 중단한다.”라는 협업 시나리오가 AHAM CHA에 정의한 디바이스 모델의 객체를 중심으로 협업 처리된다.

표 1. 디바이스 모델의 객체로 표현함
 Table. 1 Describing Objects of each Device Model

디바이스 동작	가전기기(디바이스) 모델의 객체
에어컨.시작	AirConditioner.UserControl.EnvironmentControl.StartCycle
창문.닫음	Window.UserControl.WindowControl.WindowClose
문.닫음	Door.UserControl.DoorControl.DoorClose
난방기.멈춤	Heater.UserControl.EnvironmentControl.PauseCycle

외부의 요청에 의해 자신의 제어 객체가 호출되면, 제어 객체의 협업 관계에 있는 객체도 호출된다. AirConditioner.UserControl.EnvironmentControl.StartCycle이, SetOn 메서드에 의해 호출되면, 협업 객체인 StartCycle은 InvokeAll 메서드를 함께 호출한다. InvokeAll 메서드는 co_actions에 저장되어 있는 협업 객체의 참조 정보를 StartAction메서드로 넘기고 StartAction은 협업 객체의 SetOn 메서드를 호출하도록 한다. InvokeAll 메서드의 동작 내용은 아래와 같다.

AirConditioner.UserControl.EnvironmentControl.StartCycle.StartAction(co_action[1])

AirConditioner.UserControl.EnvironmentControl.StartCycle.StartAction(co_action[2])

AirConditioner.UserControl.EnvironmentControl.StartCycle.StartAction(co_action[3])

위에서 보는 바와 같이 co_actions에 저장된 데

V. 결론

스마트 홈 기술은 기업이나 학계에서 인간의 편리한 생활을 실현하기 위해 꾸준히 연구되어 왔으며, 삶의 질이 높아짐에 따라 소비자의 수요도 새로이 창출되고 있다. 스마트 홈은 가정 내에 산재되어 있는 홈 디바이스들을 지능적으로 제어함으로써 거주자에게 서비스를 제공하게 된다. 본 논문은 스마트 홈이 제어하고자 하는 홈 디바이스들의 협업 관계와 협업관계 처리 방안을 정리하고, AHAM CHA의 디바이스 모델을 기반으로 협업관계를 처리할 수 있는 모델을 제시하였다. 제안한 협업 모델은 현재 생활 형태에서는 제한적인 디바이스 협업 형태만을 예시하여 가능성만을 보여주었지만, 향후 생활 패턴의 변화, 기술의 발달, 새로운 형태의 홈 디바이스 개발 등에 따라 스마트 홈 기술에서 많이 활용될 것으로 본다.

참고 문헌

[1] S.S. Intille. "Designing a Home of the Future", IEEE Pervasive Computing, pp. 80-86, 2002.
 [2] W. Dilger, "Decentralized Autonomous Organization of the Intelligent Home According to the Principle of the Immune System", IEEE Conference on Systems, Man,

and Cybernetics, Vol.1, pp.351-356, Oct. 1997.

- [3] In Gyu Park, "The Remote Control System for the Next Generation Air Conditioners", IEEE Transactions on Consumer Electronics, Vol. 47, No. 1, pp. 168-178, 2001.
- [4] 드림엘지, "<http://www.dreamlg.com>"
- [5] PLC 포럼 디지털 가전위원회, "Home Network Control Protocol (HNCP) PreSpec. Ver.1.0", 2002.
- [6] K. S. Lee, H. J. Choi, C. H. Kim, and S. M. Baek, "A New Protocol For Home Appliances - LnCP," IEEE ISIE 2001, pp.286-291, 2001.
- [7] 문경덕, "홈 네트워킹 미들웨어 표준화 동향," 홈네트워킹 기술 워크샵 자료집, 대한전자공학회, 2002.
- [8] AHAM, "AHAM Connected Home Appliances - Object Modeling(ANSI/AHAM CHA-2003)", 2003.
- [9] AHAM Homepage, "<http://www.aham.org>"
- [10] D.Hindus, "The Importance of Homes in Technology Research", Proc. 2nd Int'l Workshop Cooperative Buildings(CoBuild 99), Lecture Notes in Computer Science 1670, Springer-Verlag, Berlin, 1999, pp. 199-207
- [11] G. Evans, "Solving Home Automation Problems Using Artificial Intelligence Techniques", IEEE Transactions on Consumer Electronics, Vol. 37, No. 3, pp.395-400, AUG 1991.
- [12] LG전자 사이버 서비스 센터 - 제품사용설명서, "http://www.lgservice.co.kr/docs/graph/ec_man1_index.jsp"
- [13] 삼성전자 - 고객지원 - 사용설명서, "<http://www.sec.co.kr/index.jsp>"

저자 소개

김용호(Yong-ho Kim)



1993년 부산대학교 전자계산학과 (이학사)
1995년 부산대학교 대학원 전자계산학과(이학석사)

2001년 ~ 현재 부산대학교 대학원 전자계산학과 박사과정

1995년 ~ 현재 한국기계연구원 자본재정보기술그룹 선임연구원

※ 관심분야 : 인터넷응용, 홈 네트워크, 정보가전, 데이터베이스

이종환(Jong-hwan Lee)

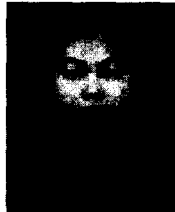


1996년 부산대학교 전자계산학과 (이학사)
1998년 부산대학교 전자계산학과 (이학석사)

2003년 부산대학교 전자계산학과(이학박사)

※ 관심분야 : 데이터베이스, 정보가전, 인터넷응용

유동회(Dong-hui Yu)



1992년 부산대학교 전자계산학과 (이학사)
1994년 부산대학교 전자계산학과 (이학석사)

2001년 부산대학교 전자계산학과(이학박사)

1997년 한국전자통신연구원 연구원

2001년~2002년 이김전자 부설 연구소장

2002년 ~ 현재 부산가톨릭대학교 컴퓨터정보공학부 조교수

※ 관심분야 : 인터넷 시각동기, 위치기반서비스, GPS, 홈네트워킹, 멀티미디어통신시스템