

VHG 기반 메타컴퓨팅 미들웨어 환경에서 사용자 중심의 가상 가전기기(UVA) 개발

박성환⁰, 한동윤, 손영성[‡], 김경석[†]
부산대학교 컴퓨터공학과⁰, 한국전자통신연구원[‡], 부산대학교 정보컴퓨터공학부[†]
{shpark⁰, dyhan, gimgs0[†]}@asadal.pusan.ac.kr, ysson[‡]@etri.re.kr

User-Centric Virtual Appliance(UVA) Development in VHG based Metacomputing Middleware Environment

Sunghwan Park⁰, Dongyun Han, Youngsung Son[‡], Kyongsok Kim[†]
Dept. of Computer Engineering, Pusan National University, Electronics and Telecommunications Research
Institute[‡], Division of Computer Science and Engineering, Pusan National University[†]

요 약

최근에 홈네트워크 분야에서 각 서비스의 특징에 따라 UPnP, HAVi, Jini 등과 같은 다양한 홈네트워크 미들웨어들이 생겨나게 되었고 이들을 통합하기 위한 연구가 진행되고 있다. 대부분의 홈네트워크 미들웨어를 통합하려고 하는 연구들은 시스템을 구축에 대한 비용이 부담된다는 것과 시스템 구축 후에 새로운 홈네트워크 미들웨어가 추가될 때 이미 구축되어 있는 시스템과의 호환이 너무 어렵다는 두 가지 문제점이 있다. 이러한 문제점들을 해결하기 위해 XML을 이용한 VHG 기반 메타컴퓨팅 미들웨어[7]를 설계하였다. 이를 XBI-Middleware라고 한다. 본 논문에서는 이 미들웨어를 바탕으로 한 환경에서 사용자가 UVAML (User-centric Virtual Appliance Markup Language)을 이용하여 간단한 시나리오 작성을 하게 한다. 이로 인해 가정 내 존재하는 가전기기들 사이의 연관성 있는 동작을 가능케 하는 사용자 중심의 가상 가전기기를 개발한다. 이를 UVA (User-centric Virtual Appliance)라고 한다. 이와 더불어 시나리오 상에 포함된 가전기기가 특정 이유로 홈네트워크에서 이탈할 경우에 나타나는 잘못된 시나리오 동작과 불필요한 시나리오 정보 저장으로 인한 스토리지 낭비와 같은 문제점들을 해결하고 그 가전기기가 다시 홈네트워크로 들어오게 되었을 경우 다시 그 시나리오의 동작을 가능케 하기 위한 시나리오 복구 방안을 제안한다.

1. 서 론

다가오는 미래에는 마이크로 프로세스가 홈 가전기기, 센서, 전등 그리고 급기야는 문이나 벽에 이르기까지 다양한 곳에 내장되게 될 것이다. 따라서 가정 내에 홈네트워크를 통해 언제 어디서든 접근가능한 다양한 가전기기가 제공되어 질 것이고 이와 함께 각각의 서비스에 맞는 다양한 홈네트워크 미들웨어가 사용되게 될 것이다. 현재 그러한 홈네트워크 미들웨어에는 UPnP (universal plug and play)[1], HAVi (home audio video interoperability)[2], Jini[3], ECHONET[4] 등이 있다. 하지만 다양한 홈네트워크 미들웨어들은 가전기기들이 물리적으로 연결되어 있다 하더라도 다른 홈네트워크 미들웨어들 간에는 상호운용을 할 수 없다. 이러한 이유로 가정 내에서 서로 다른 홈네트워크 미들웨어를 사용하는 모든 가전기기들이 호환되어서 사용하기위해 서로 다른 홈네트워크 미들웨어들이 통합되어야만 한다. 이러한 이질적인 홈네트워크 미들웨어 통합에 관련된 여러 연구들 [5], [6]이 있지만 대부분 각각의 홈네트워크 미들웨어의 모든 정보를 한 곳에 집중시키기 때문에 사용자에게 홈네트워크를 구축하기 위한 비용의 부담과 다양한 홈네트워크 미들웨어들을 통합하는 절차가 복잡하기 때문에 새로운 홈네트워크 미들웨어가 추가 될 때 재통합하는 데 어려운 점이 있었다. 이런 문제점들을 해결하기 위해 홈네트워크 상호연동을 위해 정보전달에 용이한 XML을

사용하는 VHG기반 메타컴퓨팅 미들웨어 개발[7]을 제안하였다. 이 미들웨어를 XBI-Middleware이라 부른다.

본 논문은 이전에 제안했던 XBI-Middleware 환경에서 사용자의 간단한 시나리오 작성만으로 가정 내 다양한 가전기기들이 서로 연관되어서 동작하는 가상의 가전기기를 개발하고자 한다. 이러한 가상의 가전기기를 UVA (User-centric Virtual Appliance)라고 한다. 이는 사용자가 새로운 가전기기를 구입할 필요가 없이 기존의 가전기기들의 조합으로 새로운 가상의 가전기기를 생성할 수 있기 때문에 비용에 대한 부담감이 없다. 또한 시나리오 작성 시 XBI-Middleware에서 가정 내에 존재하는 모든 가전기기와 그 가전기기의 서비스 및 동작의 리스트가 정리되어 있기 때문에 사용자는 간단히 그 리스트에서 시나리오에 필요한 가전기기와 서비스를 선택만 하면 되기 때문에 가상 가전기기 생성이 간단하다는 장점이 있다.

이러한 시나리오를 기반으로 한 UVA가 생성이 되고 난 후 UVA에 포함된 특정 가전기기가 전원이 나가거나 기기 오작동으로 인해 작동이 제대로 되지 못하는 경우 그 가전기기가 포함된 UVA는 제대로 실행되지 않게 된다. 이러한 문제를 해결하기 위해 UVA 복구 메커니즘을 제안한다.

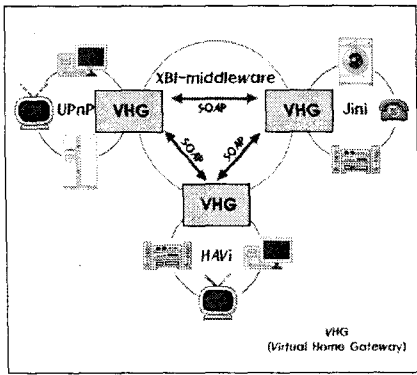
본 논문의 구성은 다음과 같다. 2장에서는 본 논문의 기반이 되는 XBI-Middleware에 대한 구조를 알아본다. 3

장에서는 XBI-Middleware상에서 사용자가 간단한 사나리오 작성을 통해 가정 내 존재하는 가전기기들 사이의 연관성 있는 동작과정을 설명한다. 4장은 UVA에 포함된 가전기기의 홈네트워크 이달로 인한 문제점을 해결하기 위해 UVA 복구를 위한 방안을 제안한다. 마지막으로 5장은 향후 연구방향을 알아보고 결론을 맺는다.

2. XBI-Middleware Architecture

2.1 XBI-Middleware

그림 1은 XBI-Middleware의 기본 구조를 나타내고 있다. UPnP, Jini 그리고 HAVi와 같은 각 홈네트워크 미들웨어 그룹에는 그룹의 홈네트워크 미들웨어와 본 논문에서 제안하는 XBI-Middleware로의 변환과 연결을 담당하는 VHG (Virtual Home Gateway)가 있다. 이 VHG가 하나의 디바이스가 되어 XBI-Middleware를 형성한다.

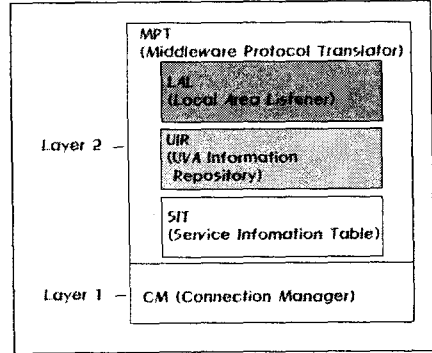


[그림 1] XBI-middleware

2.2 Virtual Home Gateway

그림 2에서 나타나는 것처럼 VHG는 SIT (Service Information Table), LAL (Local Area Listener) 그리고 새로 추가된 UIR (UVA Information Repository)과 함께 MPT (Middleware Protocol Translator)가 포함된 레이어 2와 CM (Connection Manager)을 포함하는 레이어 1로 구성된다.

MPT에서는 각각의 그룹 홈네트워크 미들웨어 (UPnP, Jini, HAVi)가 가지고 있는 Device와 Service 및 Service에 포함된 Action들의 정보를 각 가전기기가 서로 간에 메시지를 주고받을 수 있게 XML형태의 표준화된 방식으로 변환시켜준다. 그리고 레이어 2안에 포함되는 SIT는 MPT에 의해 변환되는 모든 가전기기의 Service 정보를 유지하는 정보 테이블이고, LAL은 VHG와 그 VHG가 포함된 홈네트워크 미들웨어 그룹의 가전기기들과의 연결을 담당하는 리스너로 VHG가 포함된 홈네트워크 미들웨어 그룹 내에 있는 가전기기들로부터 메시지를 받기 위해 계속 대기하고 있다. 또한 이 리스너는 본 논문에



[그림 2] Virtual Home Gateway

Middleware	Appliance	Service	Action
UPnP	HDTV	Power	On
			Off
		Channel	Up
			Down
Jini	Refrigerator	Power	On
			Off

[그림 3] Service Information Table

서 제안하는 UVA(User-Centric Virtual Appliance)를 적용할 때 그룹 내에 있는 가전기기의 상태변화를 확인하기 위해 사용된다. 자세한 사항은 3장에서 설명한다. 이 리스너는 네트워크를 이용하여 VHG가 포함된 홈네트워크 미들웨어 그룹의 가전기기들에서 VHG로 연결하기 위한 네트워크 관리자이다. 그리고 새로 추가된 UIR은 UVA가 삽입되어 시나리오를 구성할 때 사용되는 시나리오 정보 저장소이다.

CM에서는 기존의 다양한 홈네트워크 미들웨어와 같이 discovery, naming, addressing기능과 각 메시지를 서로 교환 할 수 있는 통신 전달 프로토콜이 포함된다. 본 논문에서 제안하는 방식은 XML형태로 메시지를 주고받기 위해 SOAP (Simple Object Access Protocol)[8] 프로토콜을 이용한다. 각 VHG들은 MPT에 의해 변환된 XBI-Middleware 방식으로 서로 연결되고 메시지를 주고 받는다.

SIT에서는 MPT에서 변환된 가정 내의 모든 가전기기의 서비스 정보를 테이블 형태로 유지하게 된다.

그림 3은 가정 내의 모든 가전기기의 서비스 정보를 저장하고 있는 SIT의 축약된 한 예이다.

2.3 XBI-Middleware를 적용한 홈네트워크의 초기화

홈네트워크가 초기화 되면 각 그룹 홈네트워크 미들웨어에 있는 VHG에서는 그룹 내에 있는 모든 가전기기를 검색하고 VHG내의 레이어 2에 있는 MPT를 통해 그 가전기기의 서비스 정보를 그림 4와 같이 XML 형태로 변환을 한다. 그 다음 각 그룹 홈네트워크 미들웨어의 VHG들은 SOAP 프로토콜을 이용해서 서로 메시지를 주고받아 가정 내에 있는 모든 가전기기의 서비스 정보들을 공유하고 VHG내에 있는 SIT에 테이블 형태로 저장하게 된다.

```
<Middleware type = "UPnP">
  <Appliance name = "HDTV" UID="000001">
    <Service name = "Power">
      <Action>On</Action>
      <Action>Off</Action>
    </Service>
  </Appliance>
</Middleware>
<Middleware type = "Jini">
  <Appliance name = "Refrigeration" UID="000002">
    <Service name = "Power">
      <Action>On</Action>
      <Action>Off</Action>
    </Service>
  </Appliance>
</Middleware>
```

[그림 4] MPT에 의해 변환된 SIL 파일

```
<!DOCTYPE SIL (
<ELEMENT SIL (Middleware_type*)>
<ELEMENT Middleware (Appliance*)>
<ATTLIST Middleware_type CDATA #REQUIRED>

<ELEMENT Appliance (Service*)>
<ATTLIST Appliance name CDATA #REQUIRED>
<ATTLIST Appliance UID CDATA #REQUIRED>

<ELEMENT Service (Action*)>
<ATTLIST Service name CDATA #REQUIRED>

<ELEMENT Action (#PCDATA)>
```

[그림 5] SIL DTD

그림 4는 VHG의 레이어2에 있는 MPT에 의해 가정 내 모든 가전기기의 서비스 정보를 XML형식으로 변환한 SIL(Service Information List)이고 그림 5는 SIL 문서를 기본 형식을 정의하기 위해 이번에 새로 추가된 SIL DTD 문서이다.

3. User-centric Virtual Appliance

3.1 UVA

User-centric Virtual Appliance (UVA)는 가정 내 존재

하는 가전기기를 연관성 있게 순차적으로 동작시킴으로써 능동적이고 다기능의 새로운 가전기가 생성된 것처럼 하는 가상의 가전기이다. 이것은 VHG내 SIT에 있는 가정 내의 모든 가전기기의 정보 리스트에서 사용자가 원하는 가전기기 및 서비스를 선택함으로써 생성된 다.

3.2 UVA Markup Language

UVA는 UVA Markup Language (UVAML) 파일로 정의 된다. 이 파일은 마크업 언어이기 때문에 사용자들이나 개발자들이 이 파일을 쉽게 편집할 수 있고 차 후 지원할 GUI를 사용하여 이 파일을 더욱 더 쉽게 구성할 수 있을 것이다.

다음 그림 6은 UVAML의 정의를 나타내는 기본 골격이고 그림 7은 UVAML DTD이다.

```
<UVA name = "name of UVA" operation = "True/False">
  <Appliance name = "name of appliance"
  UID="Appliance Unique Identifier">
    <Type>Appliance location on scenario</Type>
    <Service name = "name of service">
      <Action>Value of action</Action>
      <Recovery Maintenance = "True/False">
        <Old_Action>Old Value of action</Old_Action>
      </Recovery>
    </Service>

    <message_to>DESTINATION_APPLIANCE_UID</message_to>
  </Appliance>
</UVA>
```

[그림 6] UVAML definition

```
<!DOCTYPE UVA (
<ELEMENT UVA (Appliance*)>
<ATTLIST UVA name CDATA #REQUIRED>
<ATTLIST UVA Operation (True | False) "True">

<ELEMENT Appliance (Service*, message_to*)>
<ATTLIST Appliance name CDATA #REQUIRED>
<ATTLIST Appliance UID CDATA #REQUIRED>

<ELEMENT Service (Action, Recovery)*>
<ATTLIST Service name CDATA #REQUIRED>

<ELEMENT Action (#PCDATA)>
<ELEMENT Recovery (Old_Action)>
<ATTLIST Recovery Maintenance (True | False) "False">

<ELEMENT Old_Action (#PCDATA)>

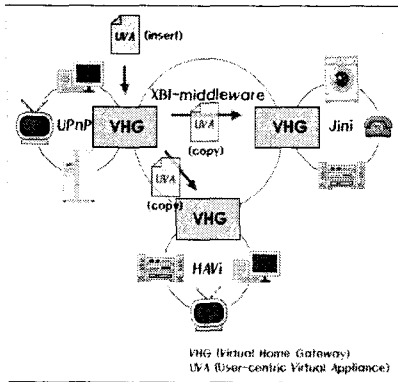
<ELEMENT message_to (#PCDATA)>
```

[그림 7] UVAML DTD

UVA는 <UVA> 태그와 <UVA> 태그안의 name속성에 UVA 이름과 Operation속성에 True와 False중 한 가지를 명시함으로 시작한다. 여기서 Operation 속성은 UVA의 동작 여부를 나타내는 것으로 True이면 UVA가 동작하고 False이면 UVA가 동작하지 않는다. 그리고 명시하지 않았을 경우 초기 값으로 True가 된다. UVA 안에는 시나리오에서 포함된 여러 개의 가전기기가 있고 <Appliance> 태그로

명시한다. <Appliance>는 가전기기를 나타내는 이름을 나타내는 name속성과 그 가전기기의 미들웨어 타입과 그 미들웨어 내 식별자를 조합한 UID의 속성이 포함된다. 이 UID를 보고 어떤 미들웨어에 포함되어 있는지, 그리고 어떤 가전기기 인지를 알 수가 있다. 각각의 가전기기는 사용되는 서비스와 이에 따른 동작과 시나리오가 진행되기 이전 상태를 나타내는 동작이 있고 <Appliance> 태그 안에 <Service> 태그와 <Service> 태그 안에 <Action> 태그와 <Recovery> 태그로 명시한다. <Service> 태그에는 서비스명을 명시하는 name 속성이 있고 <Action> 태그는 그 서비스의 동작을 나타내는 값이 들어간다. <Recovery> 태그에는 Maintenance속성 있는데 이 속성의 값은 True와 False중 한 가지를 적을 수 있다. UVA가 복구될 때 True일 경우 이전 값으로 복구할 필요가 없음을 나타내고 False일 경우 이전 값으로 복구하라는 의미이다. <Recovery> 태그 내에 포함되어 있는 <Old_Action>태그는 UVA가 실행되는 동안 동작 가전기기의 서비스가 동작하기 전의 동작 값이 들어간다. <Recovery> 부분은 4장 UVA 복구에서 자세히 다룬다.

메시지의 흐름은 <message_to> 태그로 명시하고 이는 다음에 동작할 가전기기의 UID가 들어간다. 메시지 전송 시 이 UID와 일치하는 가전기기의 서비스와 동작 정보를 포함해서 전달하게 된다.



[그림 8] UVA 삽입

3.3 UVA의 동작원리

- ① 그림 8과 같이 사용자나 개발자에 의해 생성된 UVA는 가정 내 존재하는 VHGs중 하나의 UI에 삽입되고 UVA가 먼저 삽입이 된 VHGs는 이 정보를 모든 VHGs에게 전달하여 시나리오 정보를 공유하게 된다.
- ② UVA 중에서 제일 처음에 명시된 가전기기를 관리하는 VHGs는 그것의 LAL가 그 가전기기의 상태를 체크하고 해당 조건에 해당하면 <message_to> 태그 값과 일치하는 가전기기를 관리하는 VHGs에게 그 값과 일치하는 가전기기의 서비스 및 동작 정보를 포함한 메시지를 전송하게 된다.
- ③ 다른 VHGs로부터 메시지를 받은 VHGs는 그 메시지 정

보를 자신의 MPT를 이용해 그 VHGs가 관리하는 미들웨어에 맞게 변경한 다음 실제 해당 가전기기를 동작시킨다.

- ④ 만약 UVA에 다음에 진행될 가전기기가 있을 경우 가전기기를 제어한 VHGs는 ②, ③과 같은 작업을 반복한다.

3.4 Automatic Air Conditioner UVA

이 절에서는 간단한 실제 예를 들어서 UVA가 동작하는 과정을 설명한다.

다음 그림 9는 UVAML로 작성한 Automatic Air Conditioner UVA이고 그림 10은 scenario 1을 바탕으로 한 Automatic Air Conditioner UVA의 동작을 표현한 그림이다.

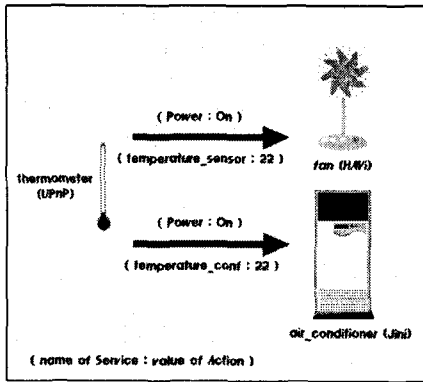
scenario 1 온도계 센스가 실온에서 28도에 이르면 에어컨과 선풍기의 전원이 켜지고 자동으로 각각 희망온도를 22도로 설정한 다음 동작한다.

```

<UVA name = "Automatic Air Conditioner" Operation="Ture">
  <Appliance name = "thermometer" UID="UPnP_0000001">
    <Type>first node</Type>
    <Service name = "sensor">
      <Action>28</Action>
      <Recovery>
        <Old_Action/>
      </Recovery>
    </Service>

    <message_to>Jini_0000005</message_to>
    <message_to>HAVi_0000009</message_to>
  </Appliance>
  <Appliance name = "air conditioner" UID="Jini_0000005">
    <Service name = "Power">
      <Action>On</Action>
      <Recovery Maintenance = "True">
        <Old_Action/>
      </Recovery>
    </Service>
    <Service name = "temperature_cont">
      <Action>22</Action>
      <Recovery Maintenance = "True">
        <Old_Action/>
      </Recovery>
    </Service>
  </Appliance>
  <Appliance name = "fan" UID="HAVi_0000009">
    <Type>end node</Type>
    <Service name = "Power">
      <Action>On</Action>
      <Recovery Maintenance = "False">
        <Old_Action/>
      </Recovery>
    </Service>
    <Service name = "temperature_sensor">
      <Action>22</Action>
      <Recovery Maintenance = "False">
        <Old_Action/>
      </Recovery>
    </Service>
  </Appliance>
</UVA>
    
```

[그림 9] Automatic Air Conditioner UVA



[그림 10] Configuration of Automatic Air Conditioner UVA

3.4.1 동작과정

- ① UPnP 미들웨어 그룹에 있는 VHГ는 thermometer (UID = UPnP_00000001)가 28도에 이르면 thermometer 내의 <message_to> 태그에 명시된 값(Jini_00000005, HVAi_00000009)과 일치하는 가전기기 (air_conditioner, fan)의 서비스와 동작 정보 메시지를 Jini와 HAVi 미들웨어 그룹에 있는 VHГ에게 보낸다.
- ② Jini와 HAVi 미들웨어 그룹에 있는 VHГ들의 MPT는 UPnP 미들웨어에 있는 VHГ에게 받은 메시지 정보를 각자의 표준에 맞는 형태로 변환시킨다.
- ③ 변환된 메시지는 실제 에어컨과 선풍기를 제어한다.

4 UVA 복구

먼저 UVA 복구 방안에 대한 설명은 다음에 정의한 용어들을 바탕으로 한다.

- off 상태의 가전기기 : 전원이 나가거나 기기 오작동으로 인해 기능을 제대로 발휘하지 못하고 홈 네트워크에서 이탈된 가전기기
- on 상태의 가전기기 : 가지고 있는 기능을 제대로 발휘하고 홈네트워크에 연결되어 있는 가전기기
- Failed UVA : off 상태의 가전기기에 의해 의도한 시나리오대로 동작하지 않거나 시나리오의 일부 분만 동작하는 UVA
- Interval Time(IT)_(sec) : 가전기기에 제어 메시지를 보내고 응답을 기다리는 시간 (10초)
- Dead Interval Time(DIT)_(sec) : 두 번의 메시지를 보내는 동안 가전기기가 응답이 없어서 off 상태의 가전기기가 되었다고 판단되는 시간 (IT*2)

UVA 생성 이후 그 UVA에서 특정 on 상태의 가전기기가 off 상태의 가전기기가 되면 그 UVA는 Failed UVA가 된

다. 이와 같은 상황이 발생하면 불필요한 정보 저장 (Failed UVA)으로 인한 스토리지 낭비와 잘못된 시나리오가 동작하는 문제점이 발생 하게 된다. 이런 문제점을 해결을 위한 복구과정을 아래에 나오는 Test UVA와 가정 1을 바탕으로 설명한다.

• Test UVA

A(UPnP, aVHG) -> B(Jini, bVHG) -> C(HAVi, cVHG)

• 가정 1

Test UVA 상에서 C 가전기기를 관리하는 cVHG는 C 가전기기에게 제어 명령을 보낸 후 C 가전기기로 부터 DIT 동안 응답이 없자 C 가전기기를 off 상태의 가전기기라고 판단한다.

Test UVA에서 A,B,C는 가전기기를 나타내고, UPnP, Jini, HAVi는 각 가전기기가 포함된 미들웨어를 나타낸다. 그리고 aVHG, bVHG, cVHG는 각 미들웨어에 포함된 VHГ이다.

4.1 UVA 복구 과정

- ① cVHG는 먼저 자신이 가지고 있는 Test UVA가 동작하지 못하게 하기 위해서 <UVA>태그에 있는 Operation 속성을 false로 수정한다.
- ② cVHG는 모든 VHГ들(aVHG, bVHG)에게 Test UVA를 UIR에서 삭제하라는 SCE_DEL 패킷을 전송한다. 이 메시지를 받은 VHГ는 Failed UVA로 인해 원하지 않은 동작을 한 가전기기들의 상태를 이전으로 되돌리기 위해 Test UVA와 관련된 모든 가전기기들을 <Recovery> 태그 하위에 위치한 <Old_Action>의 이전 정보를 이용해 다시 동작시킨다. 하지만 <Recovery> 태그 내의 Maintenance의 값이 True로 설정되어 있다면 이전 값으로 되돌리지 않는다.
- ③ UIR에서 Test UVA를 삭제한다. 단 Maintenance의 값이 True면 Test UVA를 삭제 하지 않는다.
- ④ 이후 off 상태의 가전기기가 on 상태의 가전기기로 되었을 때 그 가전기기를 관리하던 cVHG의 LAL는 그 가전기기가 살아났다는 것을 감지하고 다시 자신이 가지고 있던 Test UVA의 <UVA> 태그에 있는 operation속성을 True로 수정한 다음 수정된 Test UVA를 모든 VHГ들에게 전송함으로써 그 Test UVA가 다시 동작 가능한 상태로 되게 한다.
- ⑤의 경우 Maintenance의 값이 True일 때 Test UVA를 삭제하지 않은 것은 그 가전기기는 off 상태의 가전기기에 상관없이 계속 동작하겠다는 의미이다. 이는 사용자가 UVA 생성 시 임의로 지정할 수 있다.

이러한 복구 과정을 통해 불필요한 정보 저장(Failed UVA)으로 인해 스토리지가 낭비되는 것을 보안하고 UVA의 잘못된 동작을 방지한다. 또한 <Recovery> 태그에 Maintenance 속성을 둬으로써 UVA가 Failed UVA가 될 때 일부 가전기기는 계속 실행 되게 할 수 있다.

5. 결 론

현재 서비스의 목적에 따라 다양한 미들웨어가 가정 내에 보급 되고 앞으로 더 많은 이질적인 미들웨어가 보급 될 것이다. 이러한 문제점을 해결하기 위해 프로토콜 변환에 용이하고 정보교환을 목적으로 고안된 XML을 이용하여 가정 내에 다수의 다양한 홈네트워크 미들웨어를 통합하는 VHG 기반의 메타컴퓨팅 미들웨어 [7]를 개발하였다. 이를 XBI-Middleware라고 한다. 또한 앞으로 기술의 발달과 사용자의 요구사항에 따라 다양한 기능들을 포함한 가전기기가 제공될 것이지만 그런 다양한 기능들 때문에 가전기기의 비용이 증가 하게 될 것이고 사용자들에게 비용의 부담감을 안겨주게 될 것이다. 본 논문에서는 VHG 기반의 메타컴퓨팅 미들웨어 환경에서 사용자 중심의 가상 가전기기(UVA)를 제공함으로써 사용자들에게 UVA Markup Language(UVAML)를 이용하여 아주 간단하게 가정 내 존재하는 가전기기들을 조합하게 함으로서 사용자가 원하는 다기능의 가상 가전기기를 비용의 부담감 없이 제공한다.

또한 UVA를 삽입 시킨 후 UVA상에 포함되는 가전기기가 off 상태의 가전기기로 될 경우 발생하는 Failed UVA 문제를 해결하기 위해 복구 방법을 제공했다. 이러한 복구 방법으로 불필요한 Failed UVA를 저장함으로써 발생하는 스토리지 낭비문제를 해결하고 각 UVA는 UVA에 있는 <Recovery>정보를 이용해 Failed UVA에서 잘못된 동작을 한 가전기기들을 이전 상황으로 되돌릴 수 있기 때문에 원치 않는 동작이 된다하더라도 복구 할 수 있다.

향후 본 논문을 바탕으로 실제 구현하는 작업을 할 계획이고 본 논문은 단일 UVA에 한정해지만 다중 UVA 적용 시 발생 할 수 있는 가전기기의 충돌에 대해 충돌을 일으키는 요소들을 명시하고 이 요소들을 기반으로 홈네트워크 상에서 발생할 수 있는 충돌을 분류한다. 또한 이런 충돌 발생 시 해결할 수 있는 방안을 현재 연구 중이다.

참 고 문 헌

- [1] UPnP Forum. "Universal Plug and Play". <http://www.upnp.org>.
- [2] The Havi Organization. "Havi Version 1.1 Specification". <http://www.havi.org>.
- [3] Sun Microsystems. "JINI Architecture Specification". <http://www.sun.com/jini/>.
- [4] ECHONET CONSORTIUM, "ECHONET Specification Version 1.0", Sept. 2000.
- [5] H.Alzu, I.Sato, D.Ueno, and T.Nakajima. "A Virtual Overlay Network for Integrating Home Appliances". SAINT2002 The 2002 International Symposium on Applications and the Internet.
- [6] Bichot Guillaume, Ramaswamy Kumar, Burklin Helmut, and Stahl Thomas. "Methods for bridging a HAVi sub-network and a UPnP sub-network and

device for implementing said methods," Thomson Multimedia, 2002

- [7] 박성환, 강동훈, 손영성, 김경석, "홈네트워크 상호연동을 위한 VHG기반 메타컴퓨팅 미들웨어 개발" 한국정보과학회 2006 한국컴퓨터종합학술대회 논문집(D) p.p 280, 282 2006.6.
- [8] W3C. "Simple Object Access Protocol (SOAP) 1.1". <http://www.w3.org/TR/SOAP/>.