A DYNAMIC SERVICE COMPOSITION ENVIRONMENT ON INTEGRATED HOME NETWORK SYSTEM

Junsoo KIM
Tan Laboratory
Open source environment

Developers

App Store

Smart Phone
Open source environment (cont)

- IOS, Android
- Applications: over 500,000 apps
- For work, play and everything in between
- Market size: $200 million per month (July 2009 Mobile Metrics report)
Open source environment

Developers

App Store (SI)

Smart House
Services in networked home environment

- Ubiquitous Home Networking.
  - Centralized management of appliances
  - Providing integrated home network service
- Development of interworking solution
  - ECHONET, UPnP, DLNA, OSGi
Issues

• Un-formalized devices
  • Each house has different devices and different capability
  • We not sure that the device is exist or not

• Variable service achievement
  • ex. Home theater service, Ventilating service

• Variable service priority
  • The priority is changed by many environmental factors
The integrated home network service

Composed Services

Home Theater Service
Air Conditioning Service
Music

Networked Home Appliances
The service composition

- Appliances has been composed by one or more process as service element
  - Possible to use these processes independently as part of other services
Goal of this research

- **Realize:**
  “A Dynamic Service Composition Environment in HNS”

- **Need:**
  - Flexibility of a service on various home environment
  - Extensionality of external services
  - Adaptability to Variable device status
An approach

- Semantic understanding of the service objective
- Dynamic service priority
- Distributed service developing environment
Semantic understanding of the service objective

Make cooling the room

Service manager (agent)

Human

Cool down with open

Cool down with close

Air conditioner

Window

Curtain

Service elements
Dynamic service priority

- To provide an integrated service in home network,
  - various policies have exert an influence on a service priority
  - service priority is changed according to what a policy needs to precede.

<table>
<thead>
<tr>
<th>Policy</th>
<th>5.1ch speaker</th>
<th>Table speaker</th>
<th>Headphone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sound quality</td>
<td>High</td>
<td>Middle</td>
<td>Low</td>
</tr>
<tr>
<td>Privacy</td>
<td>Low</td>
<td>Middle</td>
<td>High</td>
</tr>
<tr>
<td>Mobility</td>
<td>Low</td>
<td>Low</td>
<td>High</td>
</tr>
</tbody>
</table>
Distributed service developing environment

SP A: temperature control

SP B: Illumination control

SP C: Home theater

SP A

SP B

SP A + B + C

HGWs
Proposed system architecture
Proposed system architecture

- **Service Composition Manager**
  - registers the service information from appliances
  - composes registered services according to a service elements that are the result from Semantic Service Representer

- **Semantic Service Representer**
  - Provides service model to compose a service on the semantics.
  - The service ontology - ”For understanding service objective”
  - The policy ontology - ”For determining service priority”

- **Link Topology Manager**
  - Determines link from a transmit device to a receive device
Service Composition Agent

1. Service Request
2. Service Query (temp, cooling, energy saving)
3. Obtain elements (Electric Fan, Window)
4. Matching with process (Electric Fan, Window)
5. Process list

Import (Name Space)
A scenario of the temperature control service

- Definition of the Temperature Sensor: a device (or process) which measures a temperature
- Definition of the Cooler: a device (or process) which makes to cool down
- Definition of the Heater: a device (or process) which makes to heat up

Step 1: set target temperature (TargetTemp)
Step 2: get current temperature (CurTemp)
Step 3: If TargetTemp > CurTemp
   Then Cool down (with cooler).
Step 4: If TargetTemp < CurTemp
   Then Heat up (with heater).
Step 5: If TargetTemp = CurTemp
   Then go to Step 2
Device description ontology model
Description of curtain

<table>
<thead>
<tr>
<th>DeviceName</th>
<th>Cutain</th>
</tr>
</thead>
</table>
| DeviceDescription1 | | Name: Cutain  
| Descriptor: URL#concealer  
| Functionality: Concealer  
| Capability: High |
| DeviceDescription2 | | Name: Cutain  
| Descriptor: URL#illuminator  
| Functionality: Illuminator  
| Capability: Variable |
| DeviceDescription3 | | Name: Cutain  
| Descriptor: URL#temp_controller  
| Functionality: Tempe  
| Capability: Variable |
Relation between policy and service
Modeling the Policy

- An earphone is evaluated by several policies, privacy, sound quality and mobility.
- According to privacy policy, the earphone is highest then other devices.
- However, the sound quality policy evaluates the earphone as less priority.

<table>
<thead>
<tr>
<th>Device Name</th>
<th>Earphone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Policy</td>
<td>Privacy</td>
</tr>
<tr>
<td></td>
<td>Priority : High</td>
</tr>
<tr>
<td>Policy</td>
<td>Sound Quality</td>
</tr>
<tr>
<td></td>
<td>Priority : Low</td>
</tr>
<tr>
<td>Policy</td>
<td>Mobility</td>
</tr>
<tr>
<td></td>
<td>Priority : High</td>
</tr>
</tbody>
</table>
Policy based dynamic service priority > Policy ontology model
Prototype

- Ontology Modeling Tool: protégé 4.1.0
- OWL-DL reasoner: FaCT++
- Defined service property: Temperature Control service
- Defined Processes: 4 Processes
  - Window, Electric Fan, Electric Radiater, Air-Con
Result

Query:
Temperature Control Service

Condition: Cooling

Policy: none
Result

Query (class expression)

TempControl and hasHeatingPossibility some HeatingPossibility

Query results

Sub classes (2)

- Aircon
- ElectricRadiator

Query:

Temperature Control Service

Condition: Heating

Policy: none
Result

**Query (class expression)**

```
TempControl and hasCoolingPossibility some CoolingPossibility and hasPowerConsume only ([HighConsumtion])
```

- **Execute**
- **Add to ontology**

**Query results**

- **Sub classes (1)**
  - Aircon

**Query:**

- **Temperature Control Service**

**Condition:**

- **Cooling**

**Policy:**

- **High Power Consumption**
Result

Query:

| TempControl and hasCoolingPossibility some CoolingPossibility and hasPowerConsume only (\{LowConsumption, NoneConsumption\}) |

Condition: Cooling

Policy: Low Power Consumption
Conclusion

• Firstly, our system provides a flexible service composition in various home environments by making the system to semantically understand an objective of a service. Moreover, it is possible to realize extendable services and priorities, since these semantic descriptions can be built separately in their own area by a name space.

• Secondly, the dynamic service priority makes the service manager determines various priority of a service element according to a required policy because various external policy definitions are providing a valuation of elements.

• Finally, we proposed the distributed service developing environment. the system is able to include an external ontology model by importing its name space. Such distributed development of a service model will improve scalability of the service model. In the near future, a distributed environment for developing home services is going to be needed by service provider