Ontology Modeling for Intelligent Domotic Environments

Dario Bonino
Politecnico di Torino, e-Lite research group
http://elite.polito.it
OUTLINE

- Domotics
- Intelligent Domotic Environments
- DogOnt
- Examples
- Conclusions
DOMOTIC ENVIRONMENTS

DOMus infoMaTICS

- (domus is the Latin for home).
- Remote lighting and appliance control have been used for years (see X10, etc.),
- Nowadays domotics is another term for the digital home, including: the networks and devices that add comfort and convenience as well as security;
- Controlling heating, air conditioning, food preparation, TVs, stereos, lights, appliances, entrance gates and security systems
ISSUES

- Many vendors on the market, each with separate, not compatible, solutions
- Different technologies (bus, powerline, wireless)
- Different protocols (KNX, MyOpen, X10, LonWorks)
- Different device features
- Different sophistication of device firmware (from simple relay to full software-based operation)
ISSUES

- Only simple automation is supported
  - Simple scenarios
  - Fixed, programmed behaviors
  - Simple comfort, security and energy saving policies

- No support for more complex interactions
  - Adaptation to user preferences
  - Context detection
  - Structural verification
  - Static and dynamic reasoning on the house state
"Environments where commercial domotic systems are extended with a low cost device (embedded PC) allowing integration and interoperation with other appliances, and supporting more sophisticated automation scenarios"
allowing integration and interoperation with other appliances, and supporting more sophisticated automation scenarios”

– Modeling environments in a semantic-rich, technology independent way
– Providing suitable querying and reasoning mechanism over the environment model
DogONT

- DogOnt [ISWCo8]
- Ontology-based formal modeling of domotic environments (OWL)
  - Technology independent device modeling
  - Operation semantics for DOG [TCEo8]
  - Interoperation semantics through relations [AMIlog9]
  - Automatic generation of inter-operation rules [TCEo9]
DogSim: A State Chart Simulator for Domotic Environments
DogONT

- 5 main facets
  - Building Environment
  - Building Thing
  - Functionality
  - State
  - Network Component
A Lamp is

- A thing (*BuildingThing*)
- It is controllable (*Controllable*)
  - (On/Off) (*OnOffFunctionality*)
- It can be either Lit or not Lit (*OnOffState*)
- It is located in a Room (*isIn Room*)
- It is connected to a domotic plant (*isA DomoticNetworkComponent*)
**Different Aspects in the Same Model**

- **Building Thing**
- **Controllable**
- **House Plant**
- **Electric System**
- **Lamp**

Connections:
- `IsIn / contains`
- `hasState`
- `hasFunctionality`

- **Functionality**: OnOff Functionality
- **State**: State
- **Discrete State**: 1
- **Building**: 1
- **Environment**: 1
- **OnOff State**: OnOff State
- **Room**: 1
- **Apartment**: 1

**Building Environment**: 1

**Control Functionality**: 1

**Functionality**: 1
<owl:Class rdf:about="#SimpleLamp">
  <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    Simple lamp that can be just turn on or turn off</rdfs:comment>
  <owl:disjointWith>
    <owl:Class rdf:about="#DimmerLamp"/>
  </owl:disjointWith>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    SimpleLamp</rdfs:label>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:someValuesFrom rdf:resource="#QueryFunctionality"/>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#hasFunctionality"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="#Lamp"/>
</owl:Class>
**Sample Room Model in DogOnt**

- **Sample Room**
  - **Lamp**
    - **OnOffState**
      - **OnCommand**
      - **OffCommand**
      - **OnOffFunctionality**
        - **hasState**
        - **hasFunctionality**
        - **generatesCmd**
  - **Switch**
    - **OnOffState**
      - **OffNotification**
      - **hasFunctionality**
      - **hasNotification**
      - **isIn**
      - **generatesCmd**
    - **OnNotification**
      - **hasNotification**
      - **hasFunctionality**
      - **isIn**
      - **generatesCmd**

**Notes:**
- DogSim: A State Chart Simulator for Domotic Environments
- 3/28/2010
**DogOnt and IDEs**

- DogOnt supports several critical features of IDEs
  - Device Modeling
  - Allows to define a **central point of configuration for real** devices
  - Abstracts from network-specific issues, exposing systems and objects as a **uniform set of devices, states and functionalities**
  - Enables **syntactic and semantic check of commands** received from external applications/devices
DogOnt and IDEs

- Features (continued…)
  - Transitive closure and Classification Reasoning allow to **decouple evolution of the model and domotic systems developments**
  - Supports the definition of top-down **inter-plant scenarios** (e.g. scenarios activated by external applications which involve devices in more than one plant)
  - Provides the basis for interoperation between plants (e.g. allowing a BTicino button to control a KNX light)
    - Frequent issue in Hospitals, Universities, Factories
RULE-BASED REASONING
INTEROPERATION

controlledObject

Dimmer Lamp

LightRegulation Functionality

OnOff Functionality

Set Command

OnOff Switch

OnOffNotification

OffNotification

OnNotification

OnOffNotification Functionality

has Functionality

has Functionality

has Command

has Command

has Command

has Command

generatesCommand

generatesCommand

generatesCommand

Set(50%)

On

Off

On

Off

Off

On

3/28/2010

DogSim: A State Chart Simulator for Domotic Environments
DogSim: A State Chart Simulator for Domotic Environments
SIMULATION – BANK DOOR

Status: READY
Currently Working on...

- Integrating Location information
- Integrating Energy information (Faisal)
- Integrating Privacy Issues (Faisal)
- ....
Questions?

Dario Bonino
dario.bonino@polito.it