

DomoBuilder: A MultiAgent Architecture for Home Automation

Andrea Addis, Giuliano Armano

IASC Group

DIEE – University of Cagliari

WOA 2010

Undicesimo Workshop Nazionale "Dagli Oggetti agli Agenti"

Rimini, 5 - 7 Settembre 2010



Introduction

- Current technologies permit people to make use of various systems able to fulfill most of their needs while being at home.
- However, their use is often not intuitive and they are also difficult to integrate.
- We propose an architectural solution called DomoBuilder, aimed at abstracting hardware (i.e., electronic devices) and software (i.e., applications, systems).

Introduction

- We put specific emphasis on the ability of simplifying human interaction with devices
- System integration is promoted, making it easier to devise complex devices that implement new behaviors while preserving ease of use.
- A case study has also been devised and implemented, to highlights the great potential of the DomoBuilder architecture.

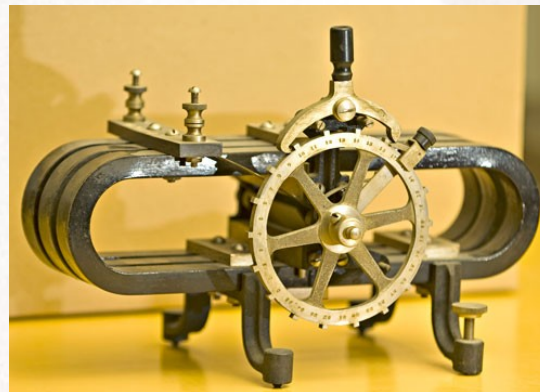
Motivations

- The new technological era provides a huge amount of innovative devices and services. However, such systems are often not intuitive, or impossible to use, without a training phase.



Motivations

- Human beings feel more comfortable with describing an object throughout its properties; in fact, problems that occur with new devices and their use are often due to an imperfect or missing description of the corresponding properties.



Motivations

- In principle a property can be read, set, or modified. However, properties can be different in their meaning and usage.
 - For instance, some are required to describe the internal state of a device without the need of exporting them to the user
 - Some are made available to the user in “read-only” mode



Motivations

- The visual development approach, in which complex components export just the features required for their usage, uses the so-called Properties, Methods, Events (PME) model.
- Encapsulating and combining those properties allows to build new interfaces and components, so that the system can be enriched with more powerful functionalities.

Motivations

- Oriented Software Engineering (AOSE) paradigm allows to embed heterogeneous devices, providing useful supports for communication, mobility persistence, pro-activeness.
- There are many evidences of the advantages in using MultiAgent Systems (MAS) for Home Automation (aka Domotics) and Ambient Intelligence (Aml).

DomoBuilder Building Blocks

- In DomoBuilder each device is intended as the building block of a system – i.e., a resource or tool that has describable properties and can encapsulate some kind of functions.
- Moreover, thanks to a centralized control, it is possible to connect devices for building complex systems (thanks to an events model).

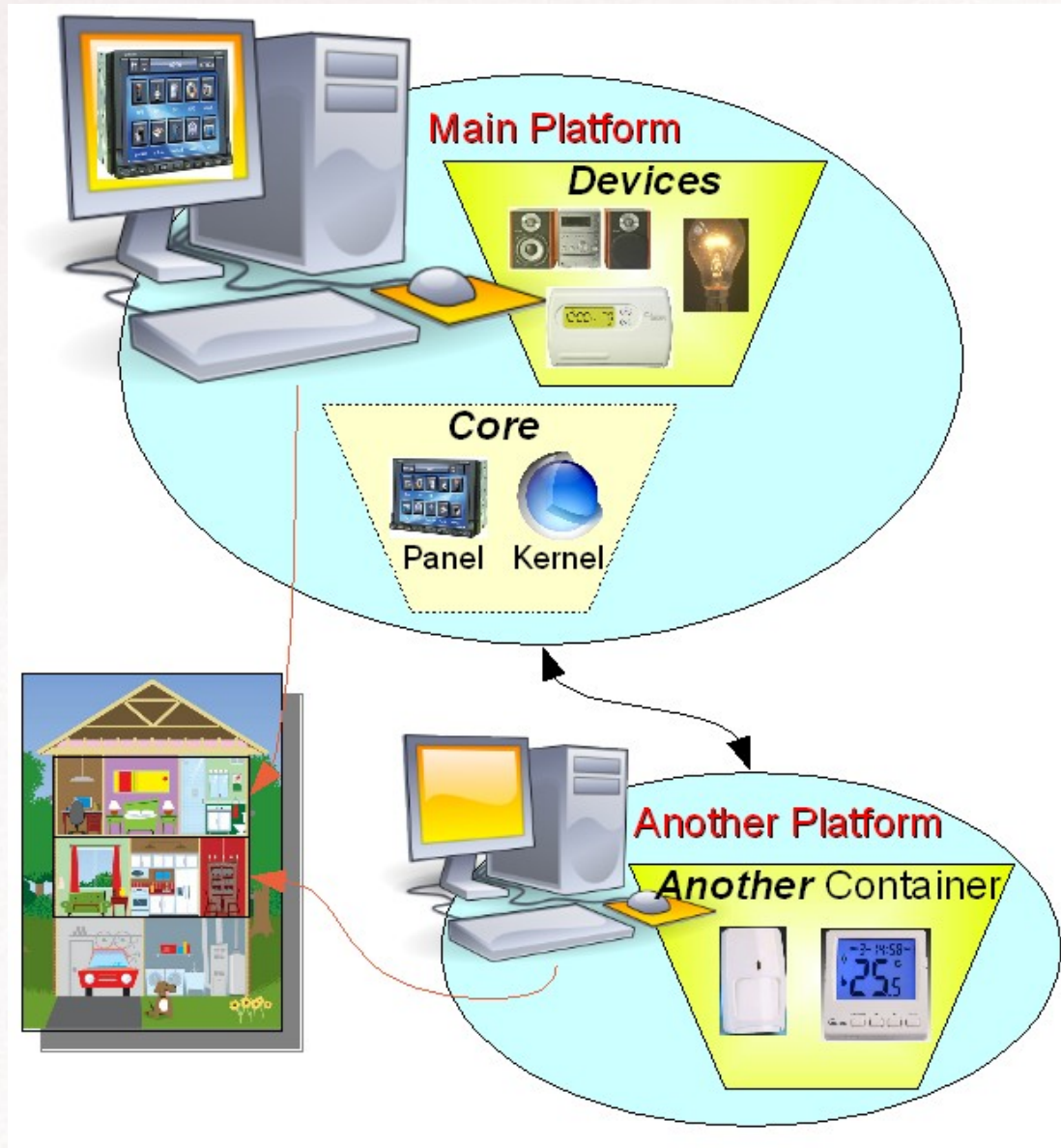
DomoBuilder Internals

- Kernel
 - The Kernel is aimed at managing (i) the life cycle of the devices populating the system and (ii) the occurring events
- Devices
 - As already pointed out, devices are agents wrapping hardware and software devices into DomoBuilder

DomoBuilder Internals

- Communication
 - Since devices are JADE agents, it is possible to communicate with them using ACL performatives according to a given ontology.
- Events
 - Events are handled by the Kernel throughout an instance of the class EventHandler

DomoBuilder Building Blocks



DomoBuilder Building Blocks

The screenshot shows the DomoBuilder - Panel v.10.02 interface. The window title is "DomoBuilder - Panel v.10.02". The menu bar includes "File", "Devices", and "Tools". The main content area is divided into several sections:

- Home**: A yellow highlighted area containing the text "Some music please". A red callout box labeled "Pyrotz Fields" points to this text.
- Ok. I am tuning the radio.**: A yellow highlighted area below the home section.
- Device Grid**: A grid of buttons for different devices: "PANEL", "Scaldabagno", "PCillo", "Parlatore", "Morettillo", "Clock", "Piantana", "Skype", and "Nabaztag". A red callout box labeled "Devices" points to the "Nabaztag" button.
- Description, and command box**: A dashed red box containing a text input field with the text "PCillo@nero:1099/JADE::domo.ultra_devices.Devicillo -> PCillo allows you to give commands in natural language".
- Device Properties**: A section with three rows: "Say" with the value "...tuning radio", "Answering" with the value "true", and "Device Properties" (labeled by a red callout box).
- Device Methods**: A section with three buttons: "Switch", "Hear", and "Reset". The "Hear" button is highlighted in blue and has a tooltip that says "What you want to ask it". A red callout box labeled "Device Methods" points to the "Hear" button.

Case Study

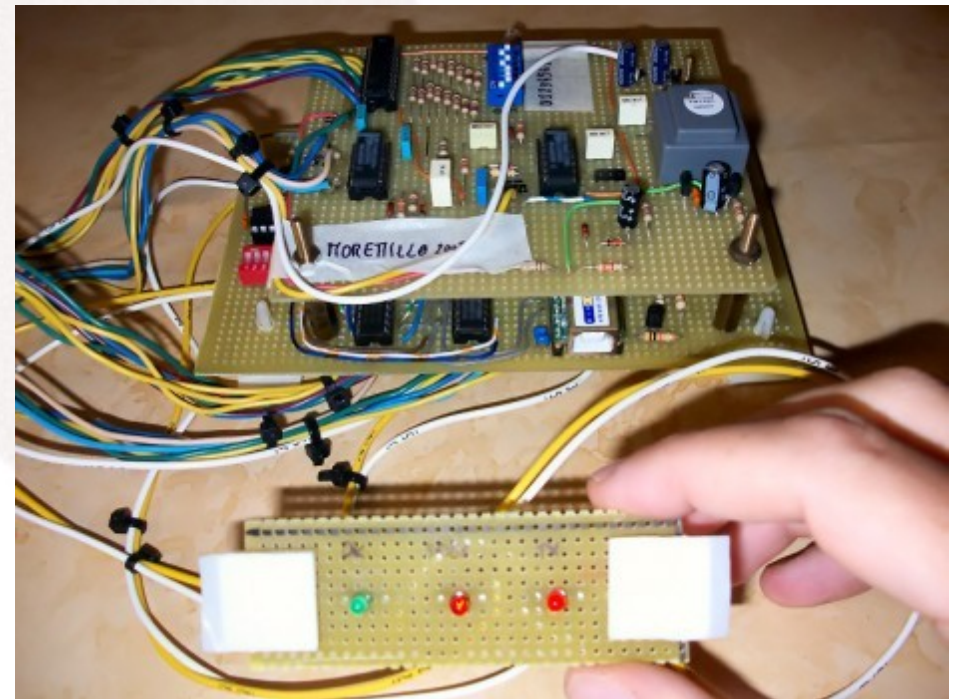
- The case study (called DomoPro) developed in the field of home automation consists of:
- Panel: A GUI for the system
- HiFi: An mp3 Reader
- Clock: A system clock with alarm
- Pyrotz: A device that can understand commands in natural language
- Movimentio: A movement sensor that exploit a computer webcam

Case Study

- Nabaztag: A device to control the Nabaztag Tag3, a Wi-Fi enabled ambient electronic device in the shape of a rabbit
- Skype: A device that interfaces the system with the Skype messenger
- Talker: A text to speech synthesizer
- MailReader: A device able to read e-mails

Case Study

- Morettillo: A device that controls the homonym general purpose hardware device
- Piantana: A device that controls a lamp
- Scaldabagno: A device that controls a house water heater



Case Study

- Building a New Device:
 - To create a device, one must extend the domo.Device DomoBuilder class, describe it, its properties and methods
 - Copy the device and ask to the kernel to add the devices (this could be made optional)

The screenshot displays a graphical user interface for configuring a device. At the top, there is a 'PANEL' header. Below it, a blue bar identifies the device as 'Clock'. A description field shows 'DESCRIPTION Clock@pacmas:1099/JADE::domo.devices.Clock -> Clock with timer and alarm function'. The main configuration area is divided into two sections: a table of properties on the left and a set of control buttons on the right.

Time			
16:56			
AlarmTime			
00:00			
Alarm			
false			

Control buttons: switchAlarm, playAlarm, setAlarmTime

Case Study

- Adding Events to DomoPro:
 - All DomoPro devices can be used in isolation. For instance, the talker could be asked to say “Hello”, the house lights can be turned on and off
 - More interesting is when DomoPro automatically undertakes decisions and executes complex behaviors (i.e., behaviors that involve different devices).

Examples

- Intrusion Detection Systems
 - Advice by email, sms if an intrusion is detected
- Personalized Domotics Applications
 - Read my email when I am back home
 - At work, ask to the house to...
- E-Health Systems
 - Turn on “the house” when awoken
 - Follow the patient and ask for its needs

Conclusions and Future Work

- Architectural solution aimed at abstracting devices
- A case study about a non-invasive, low-cost solution for home automation
- The system should take autonomous decision after learning master behavior
- The appearance must be improved

END