
Requirements Engineering II HS2016
Assignment 6: Requirements Modeling

Modeling Systems with ADORA

Fatou Ceesay
Remo Schenker
12.12.2016

Agenda

- What are the key features of the ADORA modeling language?
- How are these features supported by ADORA?
- What are the challenges related to the implementation of this tool?
- Why is it a challenge to navigate in graphical models?

What are the key features of the ADORA modeling language?

Adora is based on five principal ideas that address several issues:

- Working with **abstract objects** instead of classes
- Structuring the system being modeled with **hierarchical decomposition**
- Using an **integrated model** instead of collections of models
- Allowing users to express different parts of a specification with a **varying degree of formality**
- **Visualizing models** in context by presenting details of a model always **with an abstraction of its surrounding context**

Why abstract objects?

Problem:

Class models are **inappropriate** when:

more than one object of the **same class** and/or
collaboration between objects has to be modeled
the system should be **decomposed**

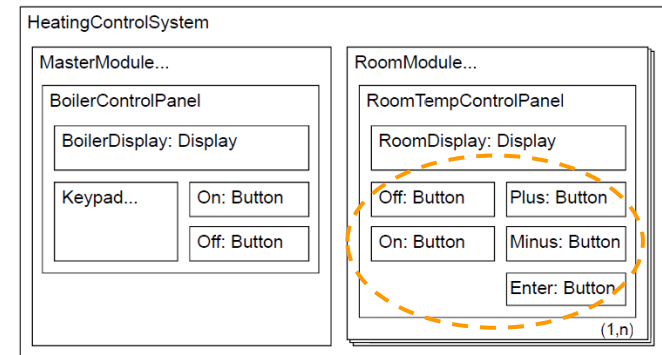
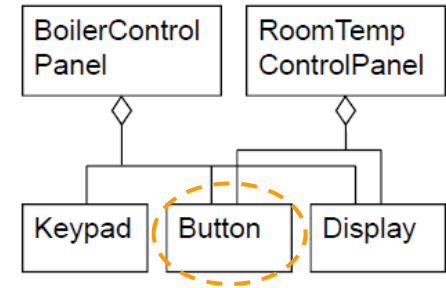
Solution:

Use:

Abstract objects -> represents a single instance

Abstract object sets -> represent a set of instances

Types -> an equivalent to classes

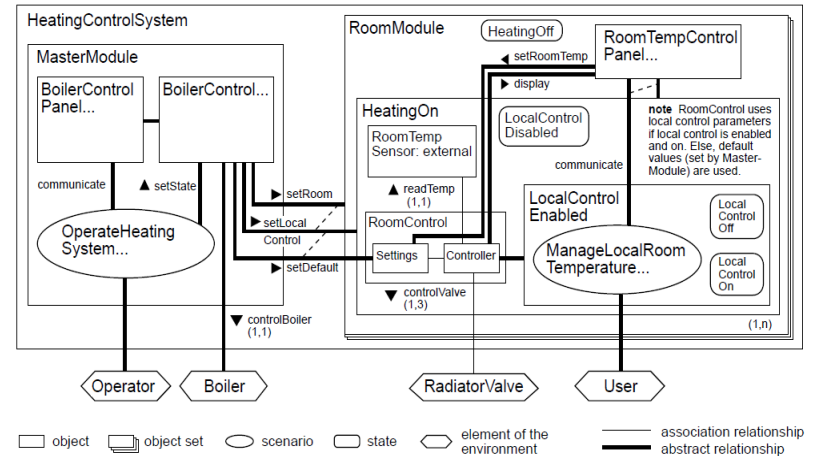
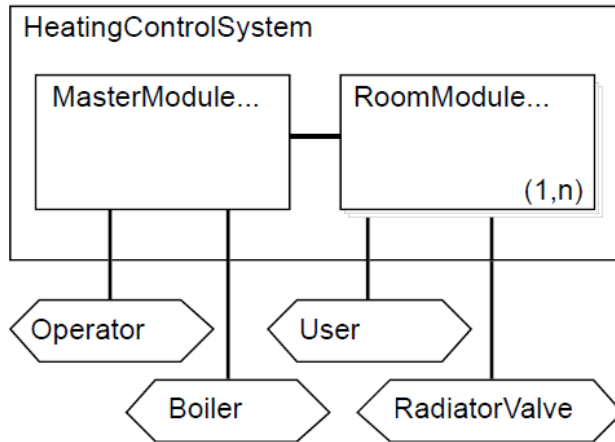


Why hierarchical decomposition?

Problem:

Good decomposition is not possible by modelling systems as **collections of models** or by using **container constructs**

Solution:



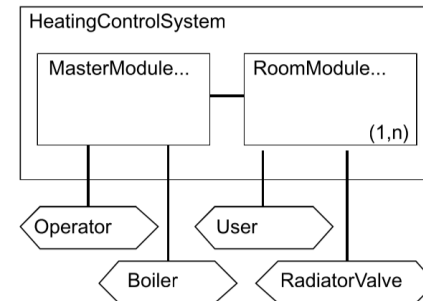
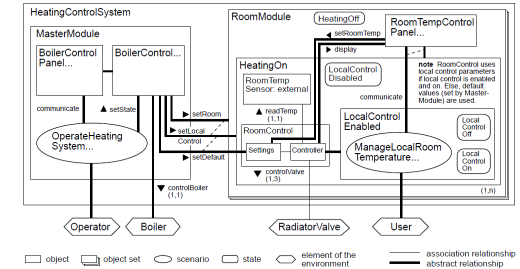
Why an integrated model?

Problem:

- Loosely coupled collections of models **make** the important issues of **integration and abstraction of concerns difficult**

Solution:

- ADORA **integrates all modeling aspects:**
 - structure, data, behavior, user interaction
 - in one coherent model



Why varying degrees of formality?

A modeling language should **allow** its users **to adapt the degree of formalism** in a specification to the **difficulty** and the **risk** of the **problem at hand**

Therefore ADORA gives the modeler the choice between:

- Informal specifications
- Formal specifications
- A mixture of both

```
object Settings
part of RoomControl
provides ActualTemp;
    "Operations to inspect / manipulate control intervals (consisting
    of start time and desired temperature), both default and user-
    defined"
requires //nothing
type
    TempIntervals is list of TempInterval;
    TempInterval is structure of (start: Time, temp: Temperature)
attribute //public attributes
    ActualTemp: Temperature
var //private attributes
    DefaultIntervals: TempIntervals; //default temperature settings
    UserSetIntervals: TempIntervals //user-defined temperature settings
syncoperation CurrentTemp (t: Time): Temperature
    pre 1 ≤ t ≤ 24*60 //minutes
    post with usi = UserSetIntervals;
        //returns desired temperature at current time t
        CurrentTemp = usi[i].temp and usi[i].start ≤ t and
        not exists j • (usi[i].start < usi[j].start ≤ t)
end CurrentTemp
syncoperation DefaultTemp (t: Time): Temperature
    "Same as CurrentTemp, but returns current default value"
end DefaultTemp
operation RevertToDefault
    post UserSetIntervals' = DefaultIntervals
end RevertToDefault
"Settings must also provide operations for setting and deleting intervals and
for browsing the currently defined intervals."
end Settings.
```

What are the challenges related to the implementation of this tool?

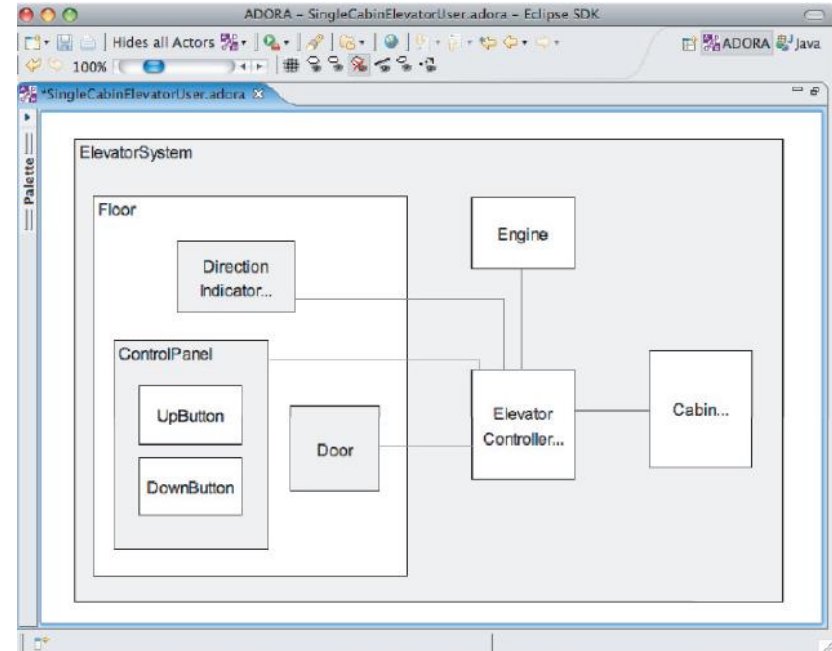
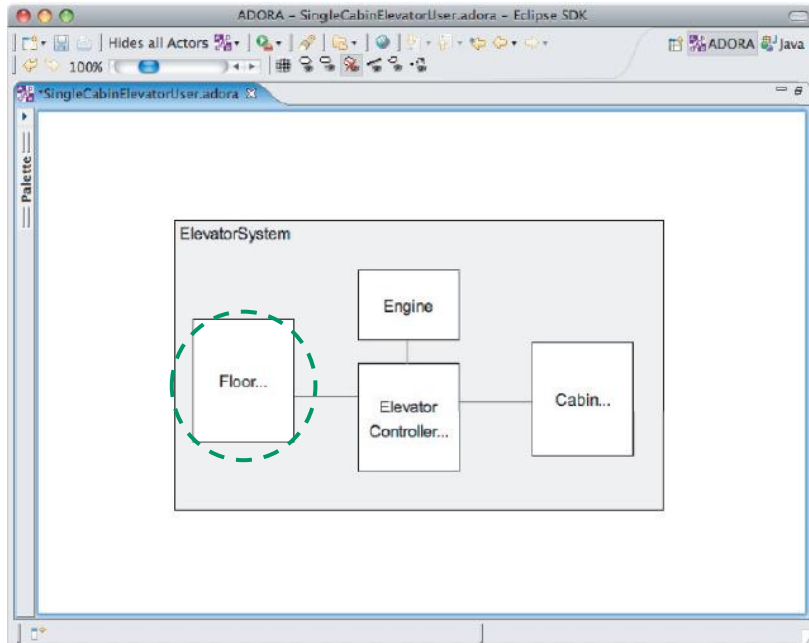
- Support orientation by visualizing as much local detail as needed without losing the global context of the focused elements
- Increase expressiveness by including semantics of the model
- Foster understandability by supporting the abstraction mechanisms of the model
- Minimize the cognitive overhead for navigation

Why is it a challenge to navigate in graphical tools?

- Scalability with increasing size of the system
- Multiple loosely coupled diagrams for different aspects of the system
- Hiding of information and separations of concern without loss of context
- Navigating back and forth between diagrams is tedious
- Zoom in a node changes size of other nodes
- Having explicit control over the layout

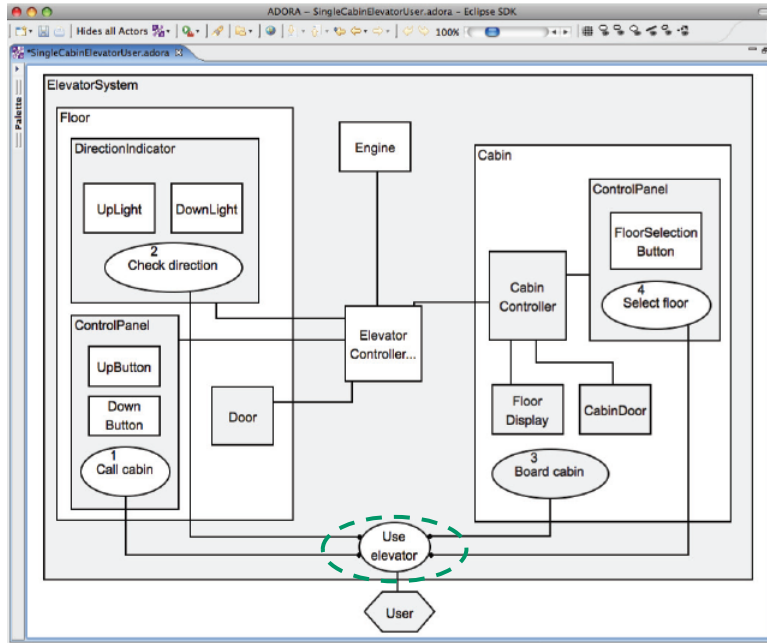
Visualising Models in Single View

Maintain surrounding Context: **FISHEYE VIEWS**

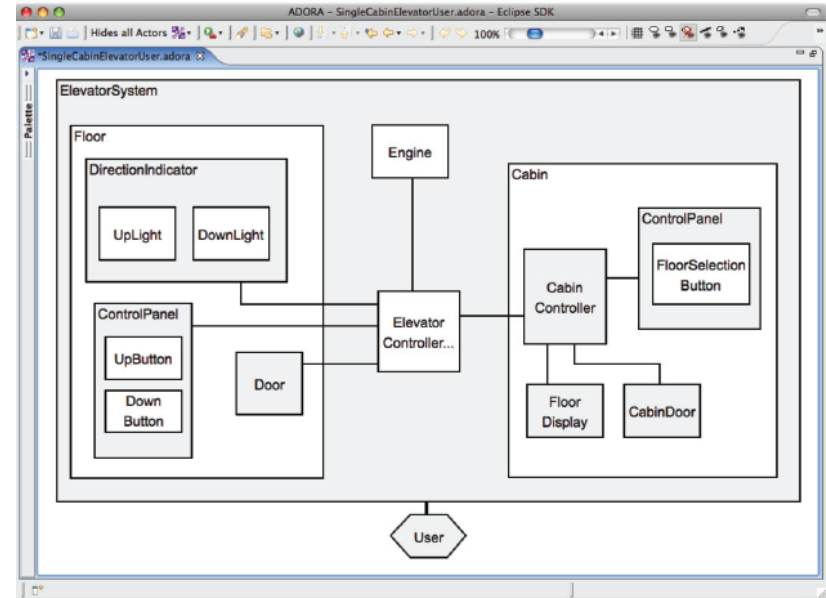


Navigation in Graphical Models: Filtered View

With User Interaction



Structural View Only



Question / Discussion

How would you compare ADORA with UML in terms of capturing model semantics?