

Modeling Systems with ADORA

Group 2

Agenda

- I. Relationship with main topic
- II. Paper I: Object-oriented modeling with Adora (2002)
- III. Paper II: Tool Support for the Navigation in Graphical Models
- IV. Conclusions

Relationship with main topic

Modeling in RE

1. We studied the role of models in software engineering
2. We discussed a set of modeling and specifications technics
3. We learned about *FlexiSketch*

Adora

1. We will learn about the object-oriented modeling tool Adora
2. How it allows to do hierarchical Navigation

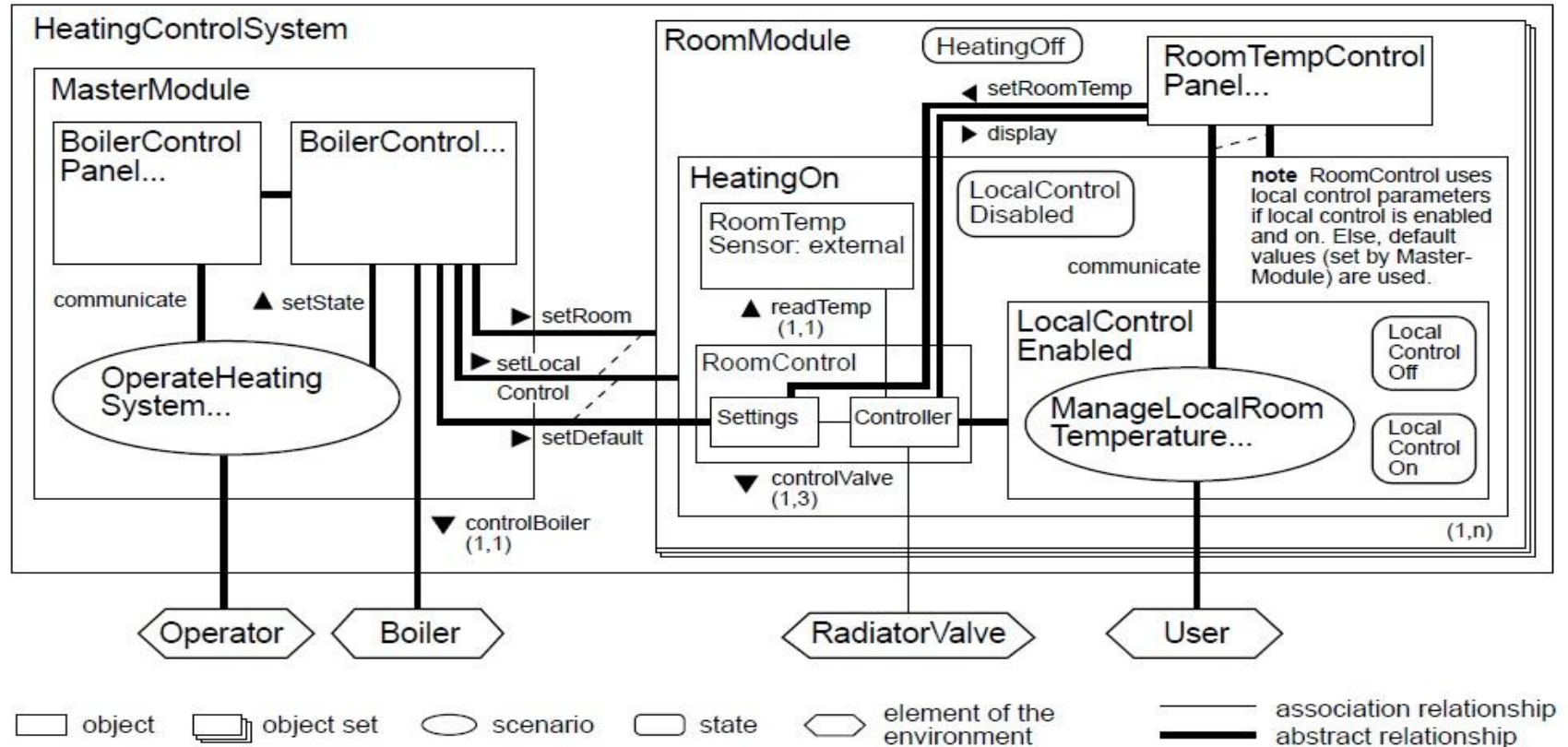
ADORA

- ADORA stands for analysis and description of requirements and architecture
- The main features are: abstract objects as the basis of the model, a systematic hierarchical decomposition of the modeled system and the integration of all aspects of the system in one coherent model

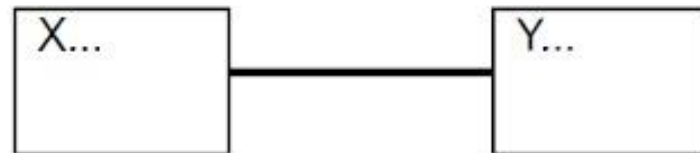
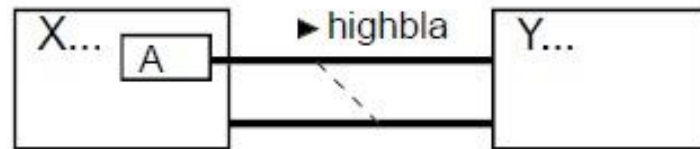
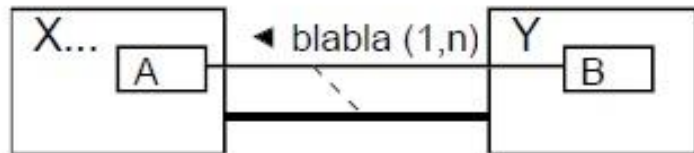
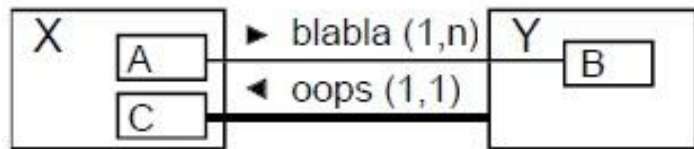
ADORA Principles

- working with abstract objects
- structuring the system being modeled with hierarchical decomposition
- using an integrated model
- 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

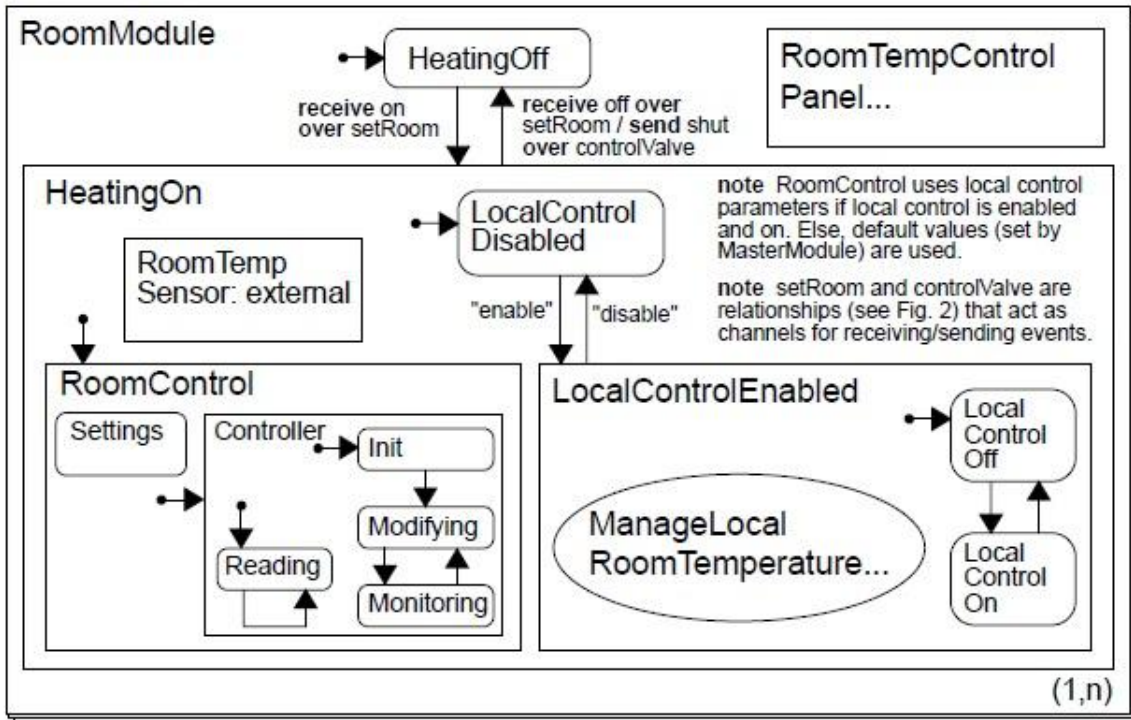
Basic hierarchical object structure



The structural view



The behavioral view



State Transition Tables for Controller

Init, Monitoring → Modifying

IN LocalControlEnabled	Y	•	Y	•
IN LocalControlEnabled.LocalControlOn	Y	N	Y	N
ActualTemp > Settings.CurrentTemp(now)	N	•	Y	•
ActualTemp < Settings.CurrentTemp(now)	Y	•	N	•
ActualTemp > Settings.DefaultTemp(now)	•	N	•	Y
ActualTemp < Settings.DefaultTemp(now)	•	Y	•	N
send open over controlValve	✓	✓		
send close over controlValve			✓	✓

Modifying → Monitoring

180 s IN Modifying	Y
--------------------	---

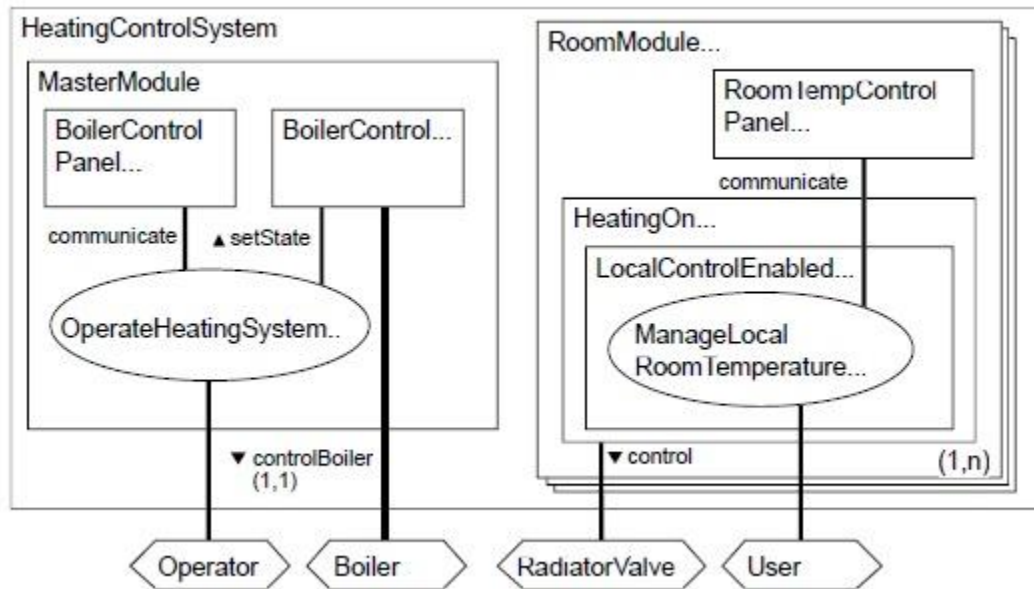
Reading → Reading

10 s IN Reading	Y
self.ReadSensorValue	✓

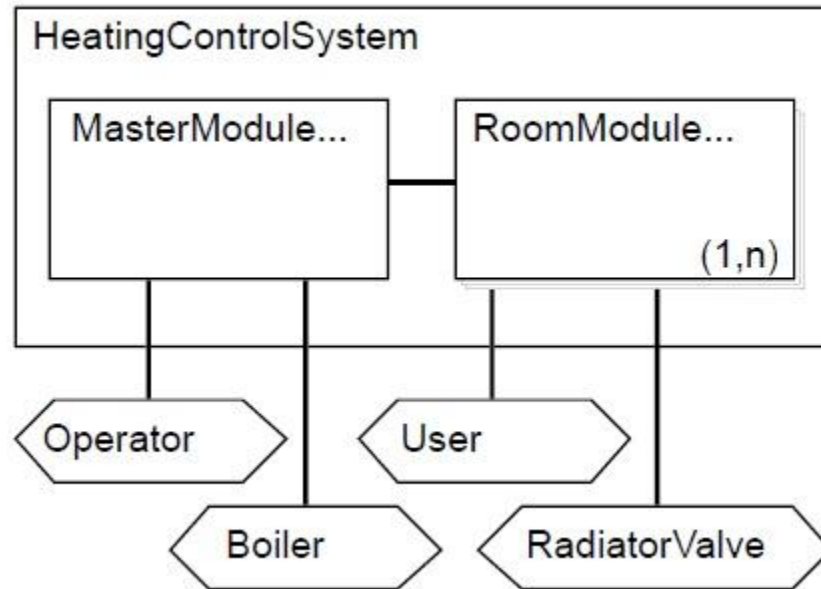
The functional view

```
object Settings
part of RoomControl
provides Actual Temp;
    "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.
```

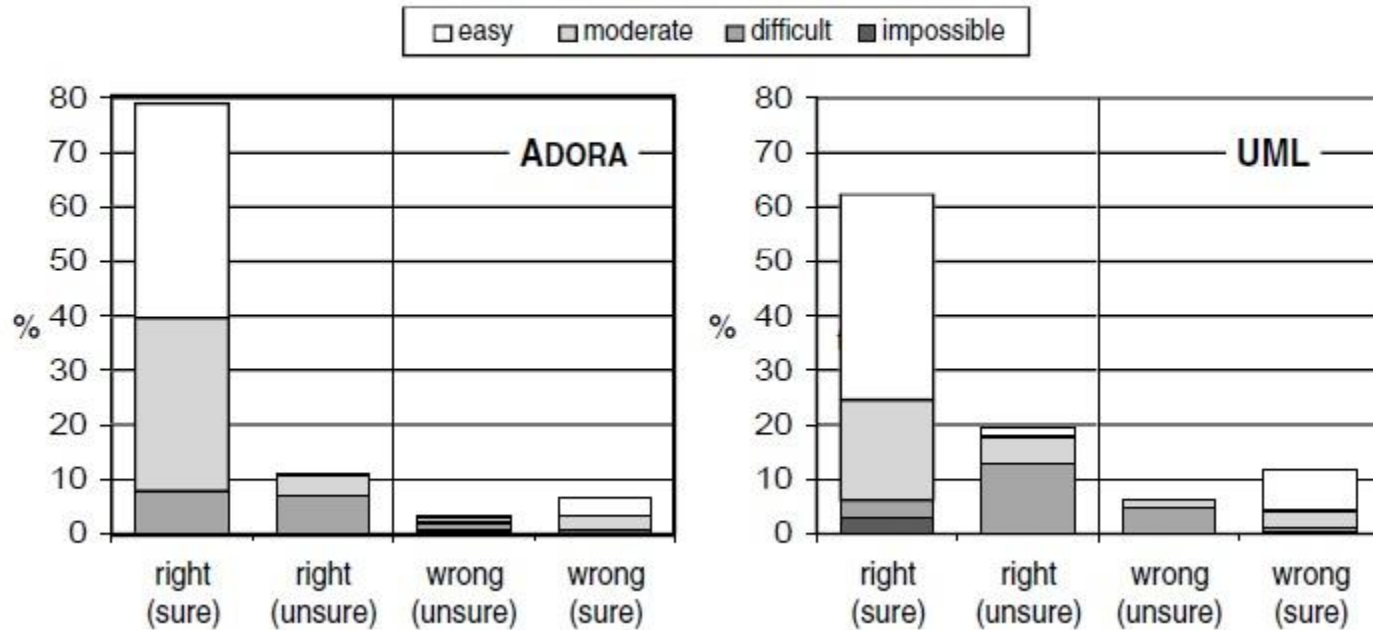
The user view



The context view



Validation of ADORA



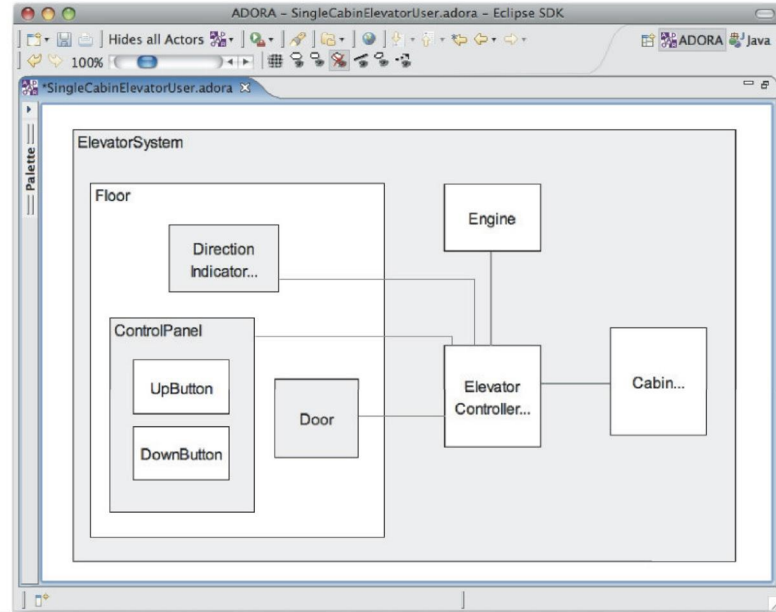
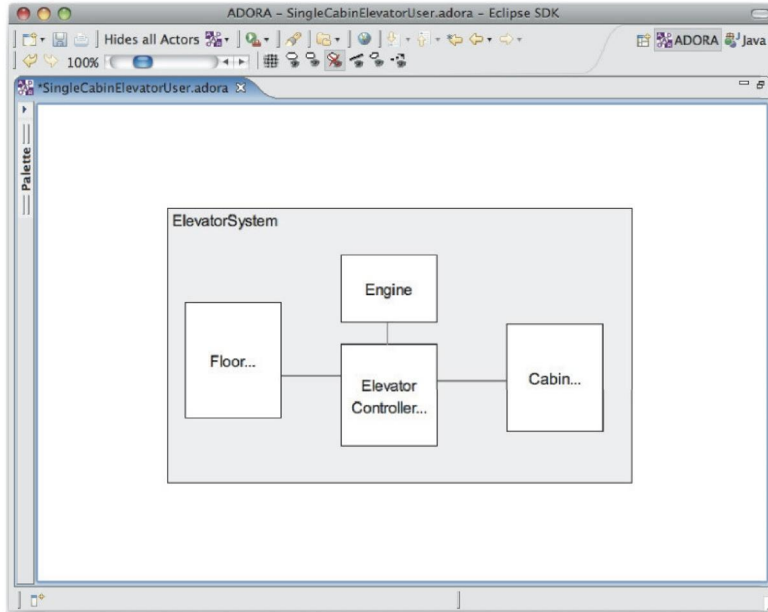
Tool Support for the Navigation in Graphical Models



Different models for:
data, functionality, behaviour,
interaction.

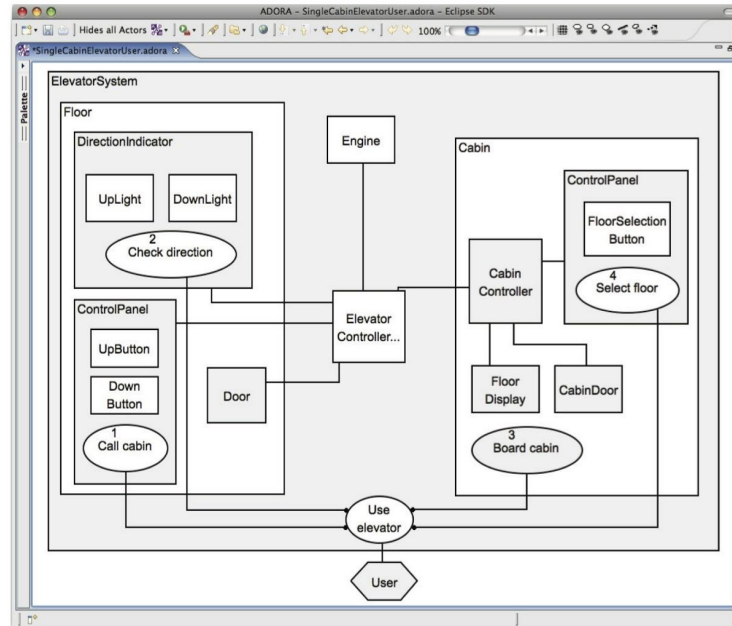
- User has to do the integration of all diagrams
- User switches back and forth between diagrams

Adora: Fisheye view I

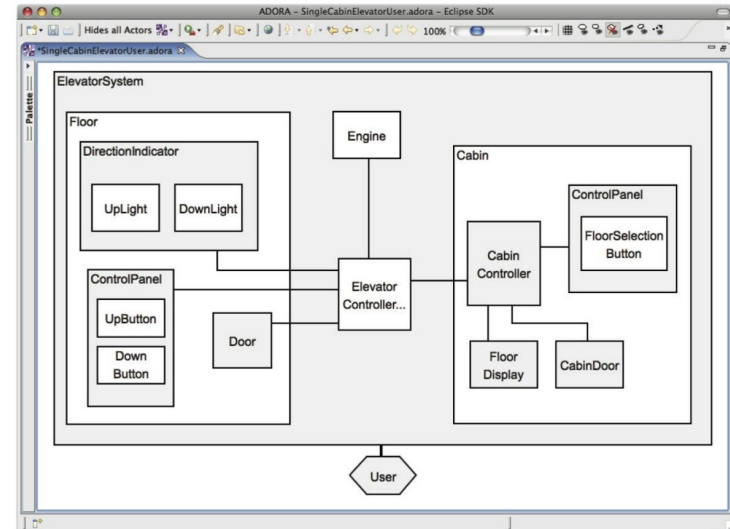


source: Reinhard2008.pdf

Adora: Fisheye view II



user view with scenarios



user view is hidden

Conclusions

- ADORA provide the integration of different views into one model.
- Coherent model reveals users from jumping between diagrams
- The fisheye zoom offered by adora helps reducing the complexity of the diagrams
- Fisheye cannot solve the problem of models growing beyond the size of the screen

Questions?

