



Integrating Variability Management and Software Architecture

Iris Groher and Rainer Weinreich



Contents

- Variability Management and Architecture – State of the Art
- Background – Architecture Management in LISA
- Variability Management in LISA
- Current and Future Work

Variability Management and Architecture

- State-of-the Art
 - Variability management and architecture development are often separated
 - Hinders traceability
 - Requires complex mappings between the models used in the different approaches
 - Danger of overengineering and model violations
- Approach
 - Integrate variability management and architecture design and development
 - Variability awareness

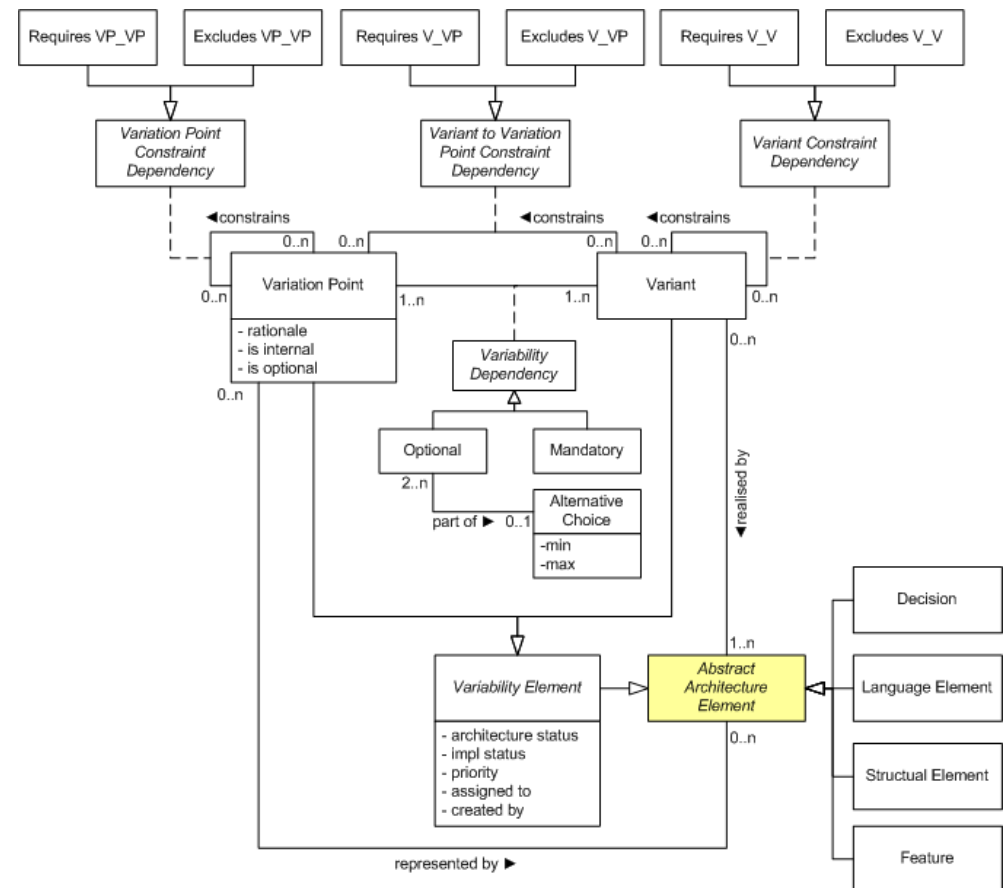


Architecture Management in LISA

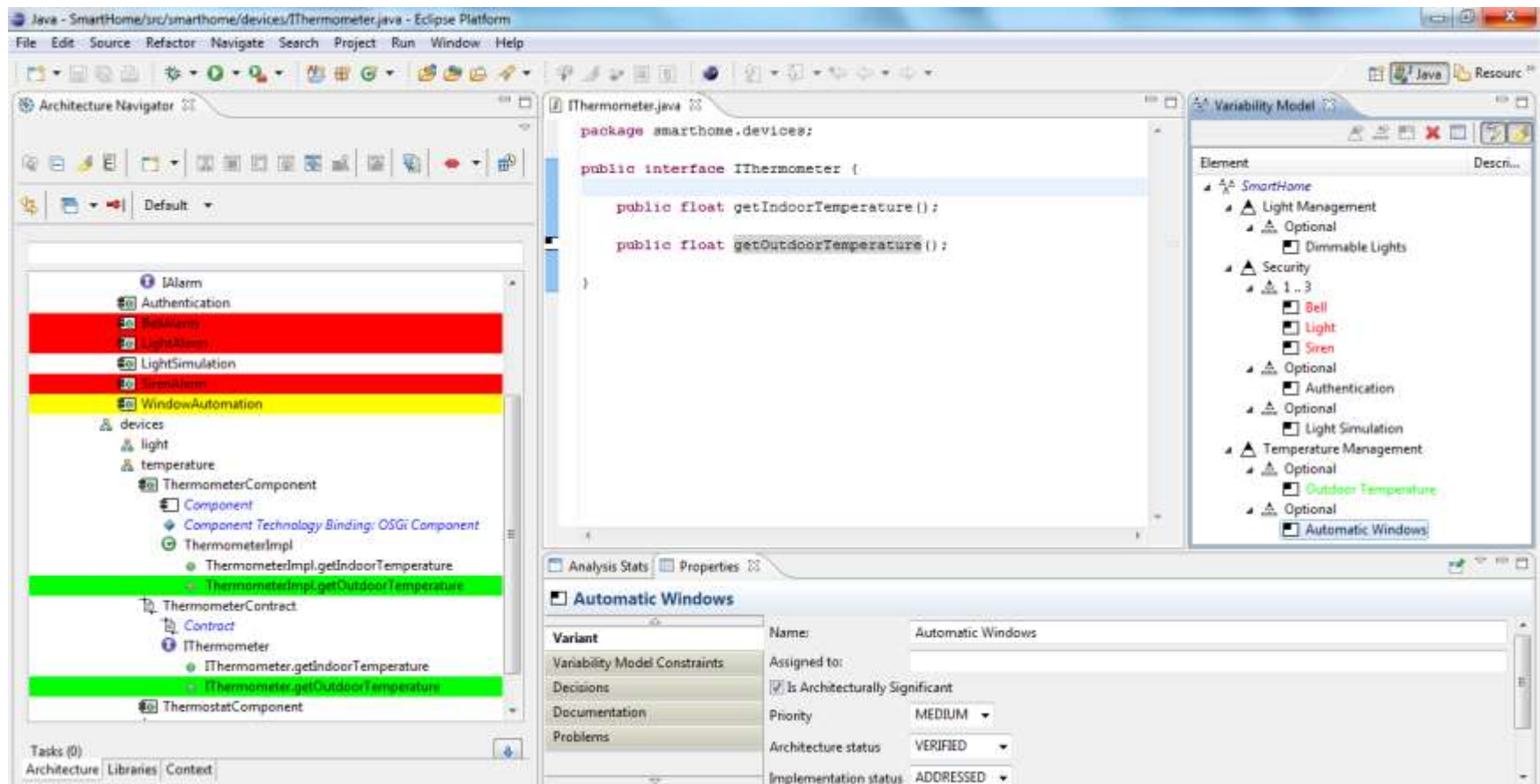
- Model and toolkit for continuous architecture management and analysis
- Single, formalized component-based architecture model
- Integration and connection of requirements, design decisions, architectural abstractions and implementation artefacts
- Creation and manipulation of architecture models, gathering and analysis of architectural knowledge, synchronization of architecture and implementation

Variability Management in LISA (1)

- Integration of the Orthogonal Variability Model (OVM)
- Feature modelling support
- Different views for working with these models
- Visualizations for variability awareness during architecture development



Variability Management in LISA (2)



The screenshot displays the Eclipse IDE interface for a Java project named 'SmartHome'. The main editor shows the source code for the `IThermometer` interface:

```
package smarthome.devices;  
  
public interface IThermometer {  
  
    public float getIndoorTemperature();  
  
    public float getOutdoorTemperature();  
  
}
```

The left-hand side features the Architecture Navigator, showing a tree view of the project's components. The `ThermometerImpl` class and its methods (`getIndoorTemperature` and `getOutdoorTemperature`) are highlighted in green. Other components like `WindowAutomation` are highlighted in yellow, and `RelAlarm` and `LightAlarm` are highlighted in red.

The right-hand side shows the Variability Model, which is a tree view of the project's variability elements. The `Automatic Windows` element is selected and highlighted in blue. The bottom-right pane shows the properties for the `Automatic Windows` element:

Property	Value
Name	Automatic Windows
Assigned to	
Is Architecturally Significant	<input checked="" type="checkbox"/>
Priority	MEDIUM
Architecture status	VERIFIED
Implementation status	ADDRESSED



Current and Future Work

- Additional visualization options
- Product derivation
- Model analysis
 - Variability-specific constraints
- Case studies

Conclusions

- Existing variability management approaches focus on product derivation
- Product line architecture development from different perspectives and viewpoints is neglected
- LISA provides integration of variability management and architecture development activities
 - Model integration – architectural and variability concepts are treated uniformly
 - Tool integration – variability management as architectural viewpoint