



Smarthome Configuration Model

Lothar Hotz* and Katharina Wolter*

*HITeC e.V., University of Hamburg, Hamburg, Germany



Contents

- Configuration model of a building automation system
- Features, components, and constraints of the domain
- Workflow of a configuration process
- Complex configuration problems



Goals

How to model features and components?

Feature and system models and their relations

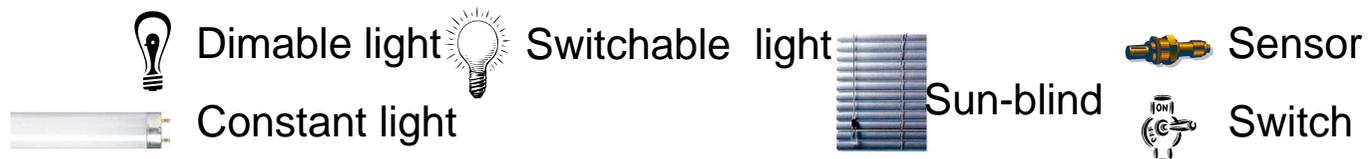
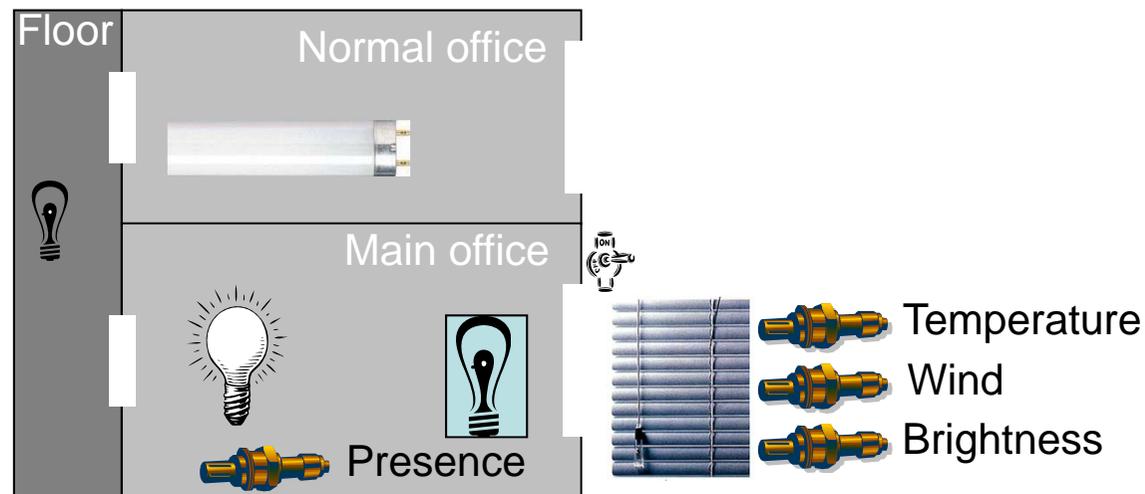
How to model complex restrictions?

Various constraint types

How to model what to configure when?

Workflow of the configuration process

Components of a Building Automation System





Features

Features are characteristics of products that are visible to the customers (Kang et al., 1990).

Features are related to components.
The components *realize* the feature.

Examples for the building automation system:

- Detect the presence of a person in a room.
- Control the sun-blinds depending on wind or temperature changes.
- Allow switchable and dimmable light controls.



Feature Model and System Model

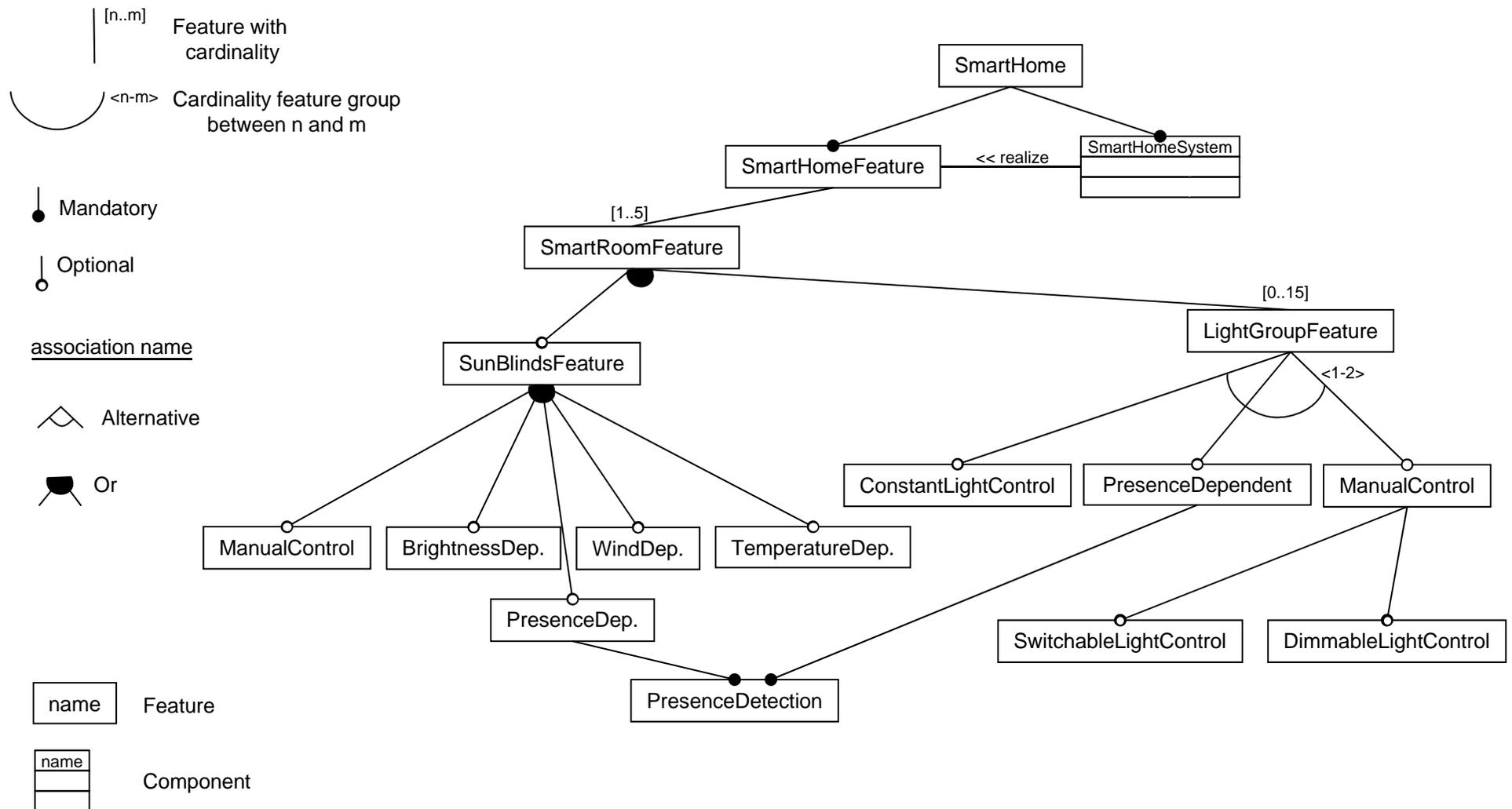
The feature model represents all features of a system.

The system model represents all components of a system.

Both are related with the realize relationship.



Feature Model Example





New Feature Modeling Facilities

Feature with cardinality

Feature group: Collection of features

Feature group with cardinality specifies how many features are in the collection.

Associations for relating the feature model to the system model



Feature Model Constraints

Not all combinations of features are realizable.
Feature model constraints reduce the possible combinations of features by specifying incompatibilities.

Incompatibility

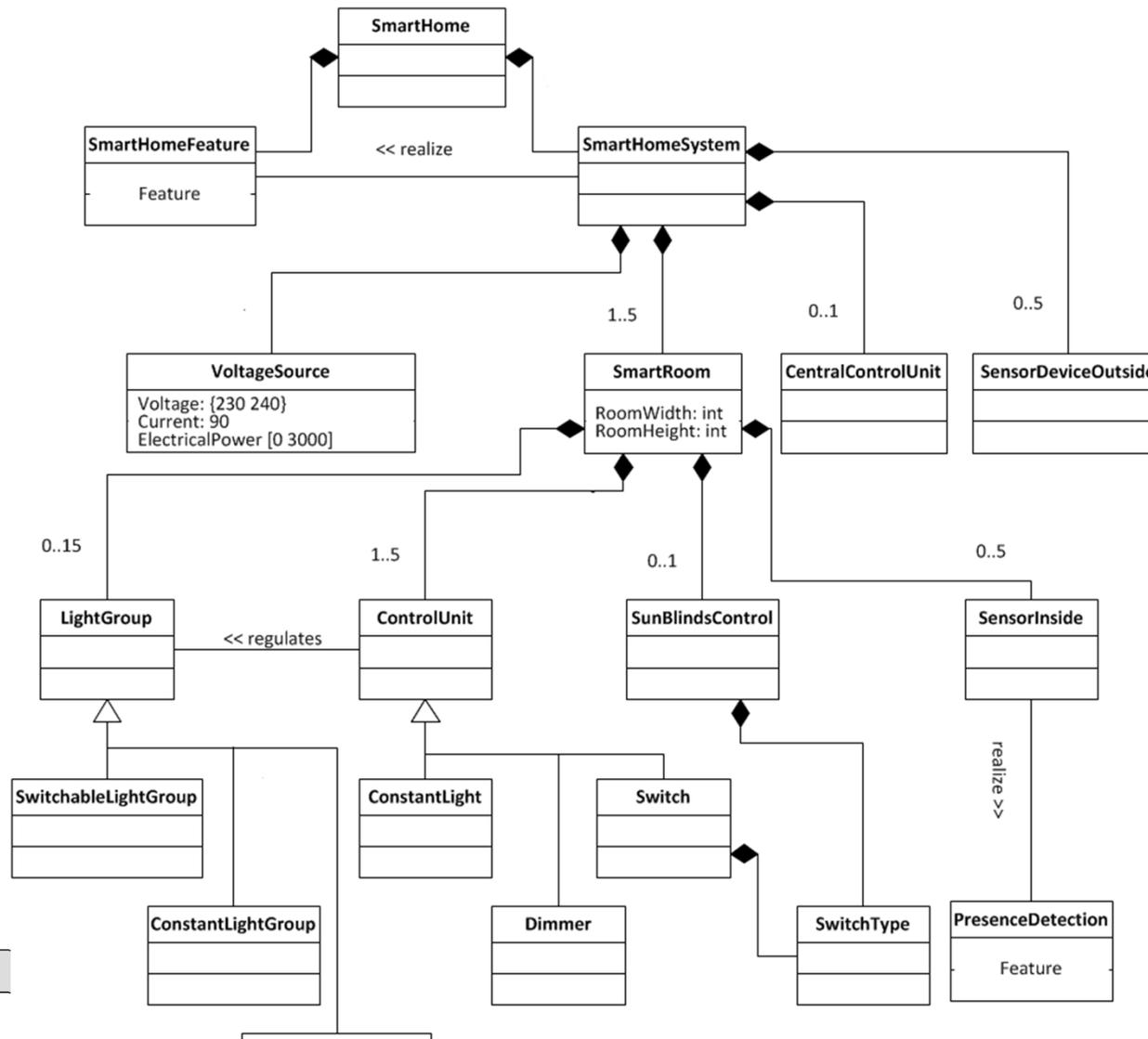
- A light group of type *SwitchableLightControl* is incompatible with a light group of type *DimmableLightControl* in the same room.

Constraint for sharing classes

- Each *LightGroupFeature* with a *PresenceDependent* requires a distinct *PresenceDetection*. Thus, the feature *PresenceDetection* is only shared for classes and not for instances.



System Model Example





Modeling Facilities of the System Model

Classes

Instances

Part-of

Specialization

Attributes

Requires

Incompatible-with

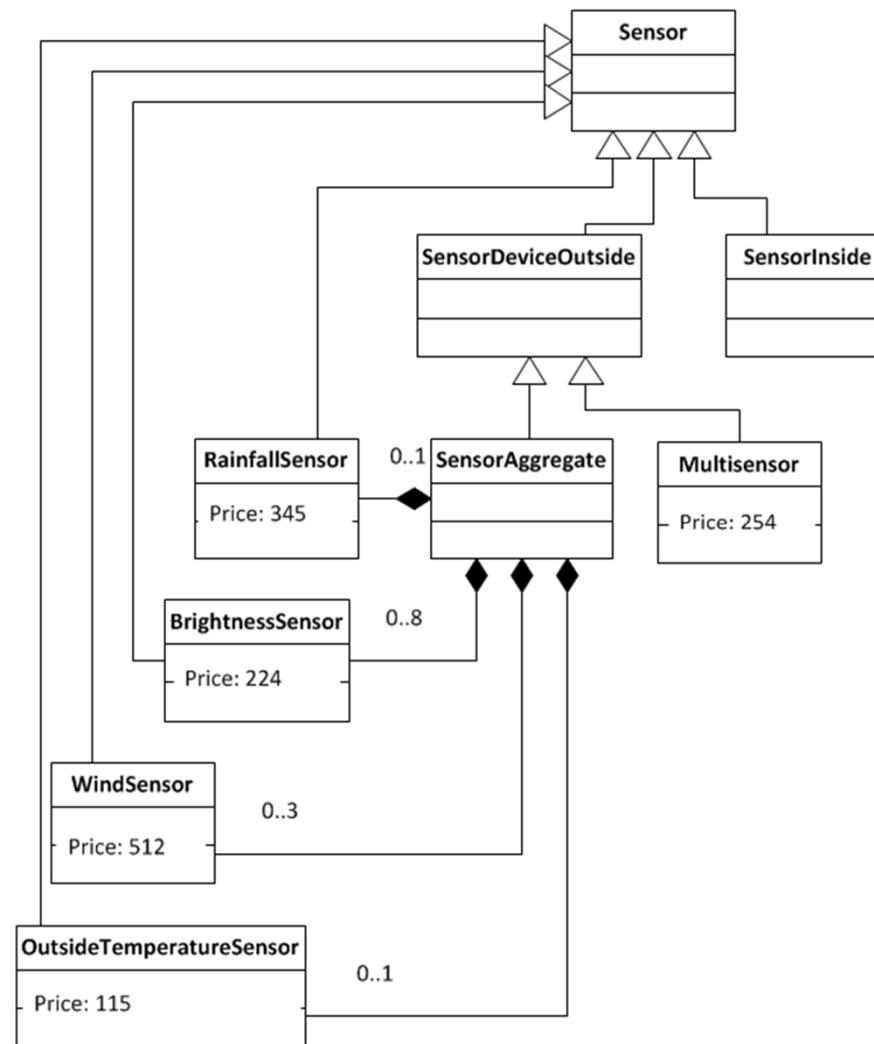
Compatible

Shared class

Connections/Associations



Separate Sensors vs. Multisensor



Algebraic Constraints

Algebraic constraint for expressing a physical restriction. Example: ElectricalPower

Algebraic constraint for a compositional relation.
Example: Computation of the needed number of PresenceDetectors

Algebraic constraint for a varying number of attributes.
Example: Price of a previously unknown number of components.



Constraint between Feature and System Model

Representing the relationship between features and components.

Feature Selection			Required components				
SunBlindsFeature			SensorOutside of the SmartHome				
Wind Dependent	Brightness Dependent	Temperature Dependent	SensorDevice-A	WindSensor	BrightnessSensor	TemperaturSensor	MultiSensor-A
yes	yes	yes	yes	yes	yes	yes	no
yes	yes	no	yes	yes	yes	no	no
yes	no	yes	yes	yes	no	yes	no
yes	no	no	yes	yes	no	no	no
no	yes	yes	no	no	no	no	yes
no	yes	no	no	no	no	no	yes
no	no	yes	no	no	no	no	yes
no	no	no	no	no	no	no	no

Configuration Workflow

Configuration decision:

- Selection of a feature
- Selection of a component
- Setting a component's attribute

Dependencies between configuration decisions:

- First features than components.

A configuration workflow specifies an order of configuration decisions to ensure an effective configuration process.



Workflow Terminology

Configuration activity describes the configuration of a certain subsystem, e.g., the configuration of a lighting system.

A configuration activity composes *steps*.

Each step represents *a configuration decision*.

Configuration process composes a number of configuration activities.

Software Process Engineering Metamodel Specification (SPEM)

SPEM can be used to model a configuration process.

An activity describes a piece of work performed by one process role. An activity consists of one or more atomic elements called steps.

The activity *Configure ControlUnit* is assigned to the class *ControlUnit*. Related decisions:

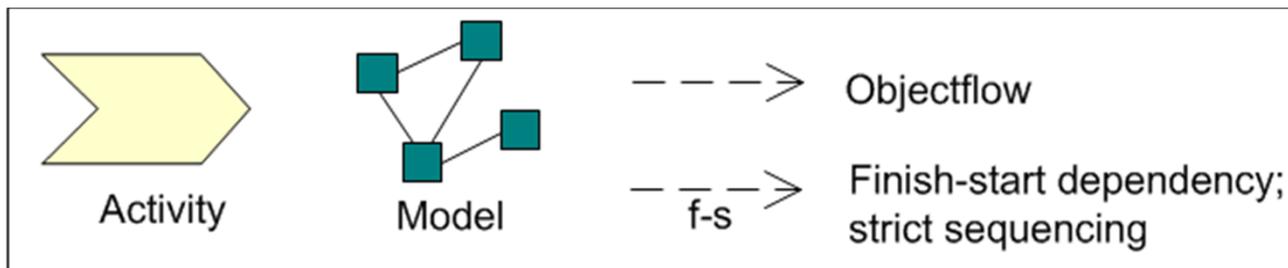
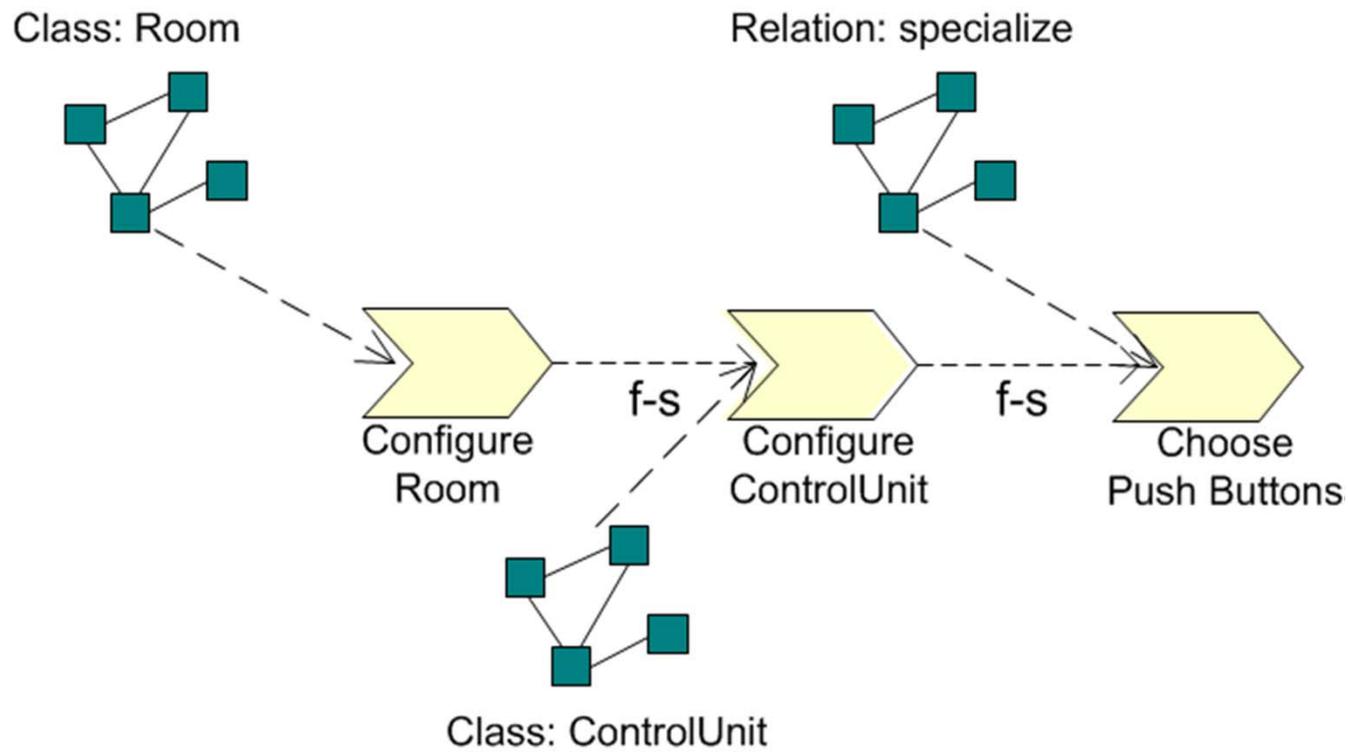
- configure the association *regulates* to a LightGroup
- make sure that the ControlUnit is part of a SmartRoom,
- determine the values of the inherited attributes PowerConsumption and Price



Software Process Engineering Metamodel Specification (SPEM)

A *model* is a specific kind of work product. A work product is anything produced, consumed or modified by a process.

If activity B has a *finish-start dependency* with respect to activity A, then B can only start after A has finished.





Characteristics of Complex Configuration Problems

- Separation of features and realizing components through a feature model and a system model:
 - The feature model provides a user view of the system.
 - In the system model, decisions between distinct devices or device categories have to be made to realize the features.
 - A feature model might be simple while the system model is more complex because of the internal structure of the components.
 - The feature model and the system model can easily be represented with the modeling facilities of a typical configuration system.



Characteristics of Complex Configuration Problems

- Modeling of a workflow for the configuration process
 - Configuration activities depend on each other.
 - Configuration activities sometimes follow a specific order.
 - Configuration process modeling belongs to a complex configuration model.
 - Configuration activities are often implemented in the configurator's user interface, i.e., cannot be represented with the configuration language of a configuration system.



Exercises

1. What are the differences between requirements, features, and components?
2. Specify the algebraic constraint
 $\text{Power} = \text{Voltage} * \text{Current}$ in OCR.
3. Elaborate on the need of configuration process models.
4. Show the SPEM model for the dependency:
“The necessary network technology depends on the number and type of the needed devices.”



Thank You!

References (1)

- (1) Czarnecki, K., Helsen, S., Eisenecker, U., 2005. Formalizing cardinality-based feature models and their specialization. *Software Process: Improvement and Practice* 10 (1), 7–29.
- (2) Hotz, L., Günter, A., 2014. KONWERK. In: Felfernig, A., Hotz, L., Bagley, C., Tiihonen, J. (Eds.), *Knowledge-based Configuration – From Research to Business Cases*. Morgan Kaufmann Publishers, Waltham, MA, pp. 281–295 (Chapter 24).
- (3) Hotz, L., Wolter, K., Krebs, T., Deelstra, S., Sinnema, M., Nijhuis, J., MacGregor, J., 2006. *Configuration in Industrial Product Families – The ConIPF Methodology*. IOS Press, Berlin.
- (4) Hotz, L., Felfernig, A., Stumptner, M., Ryabokon, A., Bagley, C., Wolter, K., 2014. Configuration knowledge representation and reasoning. In: Felfernig, A., Hotz, L., Bagley, C., Tiihonen, J. (Eds.), *Knowledge-based Configuration – From Research to Business Cases*. Morgan Kaufmann Publishers, Waltham, MA, pp. 41–72 (Chapter 6).
- (5) Kang, K., Cohen, S., Hess, J., Novak, W., Peterson, S., 1990. *Feature-Oriented Domain Analysis Feasibility Study (FODA)*. Technical Report CMU/SEI-90-TR-021, Carnegie Mellon University, Software Engineering Institute, Pittsburgh, PA.
- (6) Wolter, K., 2003. *Supporting the orientation and navigation of users in the process of configuring complex products (Master's thesis)*. Universität Hamburg, Fachbereich Informatik, Hamburg, Germany (in German: *Unterstützung der Orientierung und Navigation von Benutzern im Prozess der Konfiguration von komplexen, variantenreichen Produkten*).