Concern-Driven Development

Gunter Mussbacher / Jörg Kienzle
Software Engineering Laboratory
Department of Electrical and Computer Engineering / School of Computer Science
McGill University
Montreal, Canada
Email: {Gunter.Mussbacher, Joerg.Kienzle}@mcgill.ca

The Vision

Some years from now…
a software engineer is tasked to build a new application…

She broadly identifies domain-specific and generic concerns

She can access a virtual software engineering bookshelf filled with generic reusable concern units

- Complete set of models from requirements to implementation
- Interactive guidelines
  - Variations of the concern + their implications on system goals
  - Composition instructions

A concern specialist is on call, if help is needed

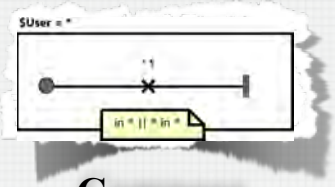
EXAMPLE

APPLICATION

Select Desired Features
taking tradeoffs into consideration

Compose

Feature level: F1 reuses F2, G1 impacts G2



Compose

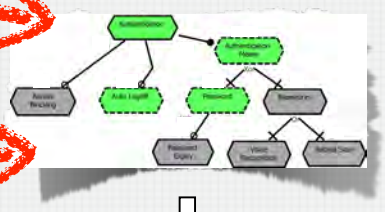
Workflow level:
- When an input reaches the system

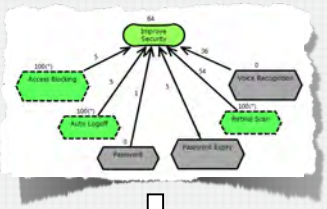
aspect bCMSDomainWithAuthentication extends bCMSDomain depends on UserCredentials	
structural view	
Instantiations: UserCredentials:	IAuthenticatable → Coordinator, Driver

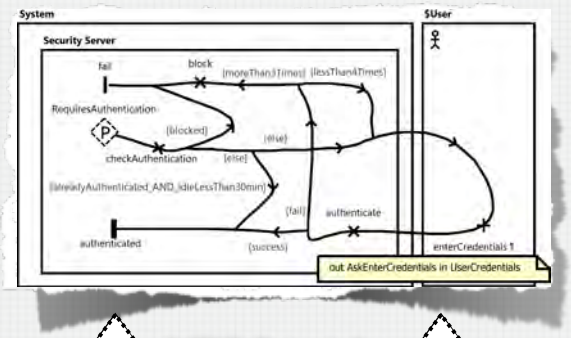
Compose

Design level:
- Affected objects (e.g., Coordinator, First Aid Personnel, Drivers in a Crisis Management System)

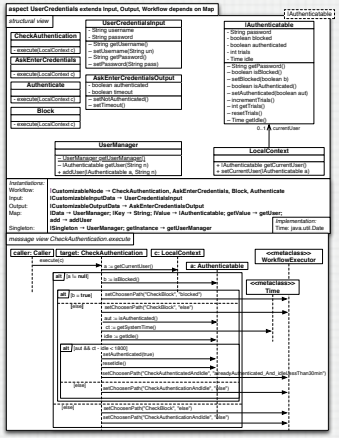
Generic Authentication Concern

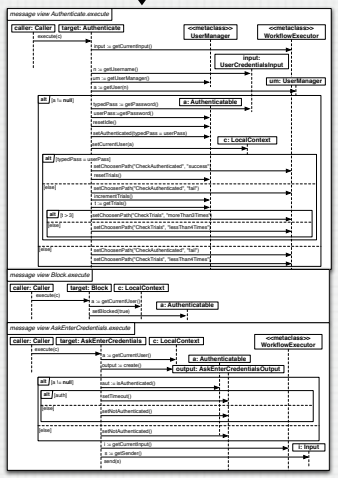






MDE Principles

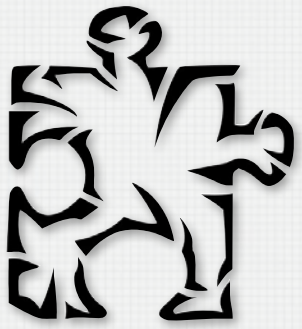




1st Workshop on Concern-Oriented Reuse © Gunter Mussbacher and Jörg Kienzle 2014

Concern Characteristics (1)

- Encapsulates composable reusable models of a generic concern (i.e., not product-specific) with well-defined interfaces for all models



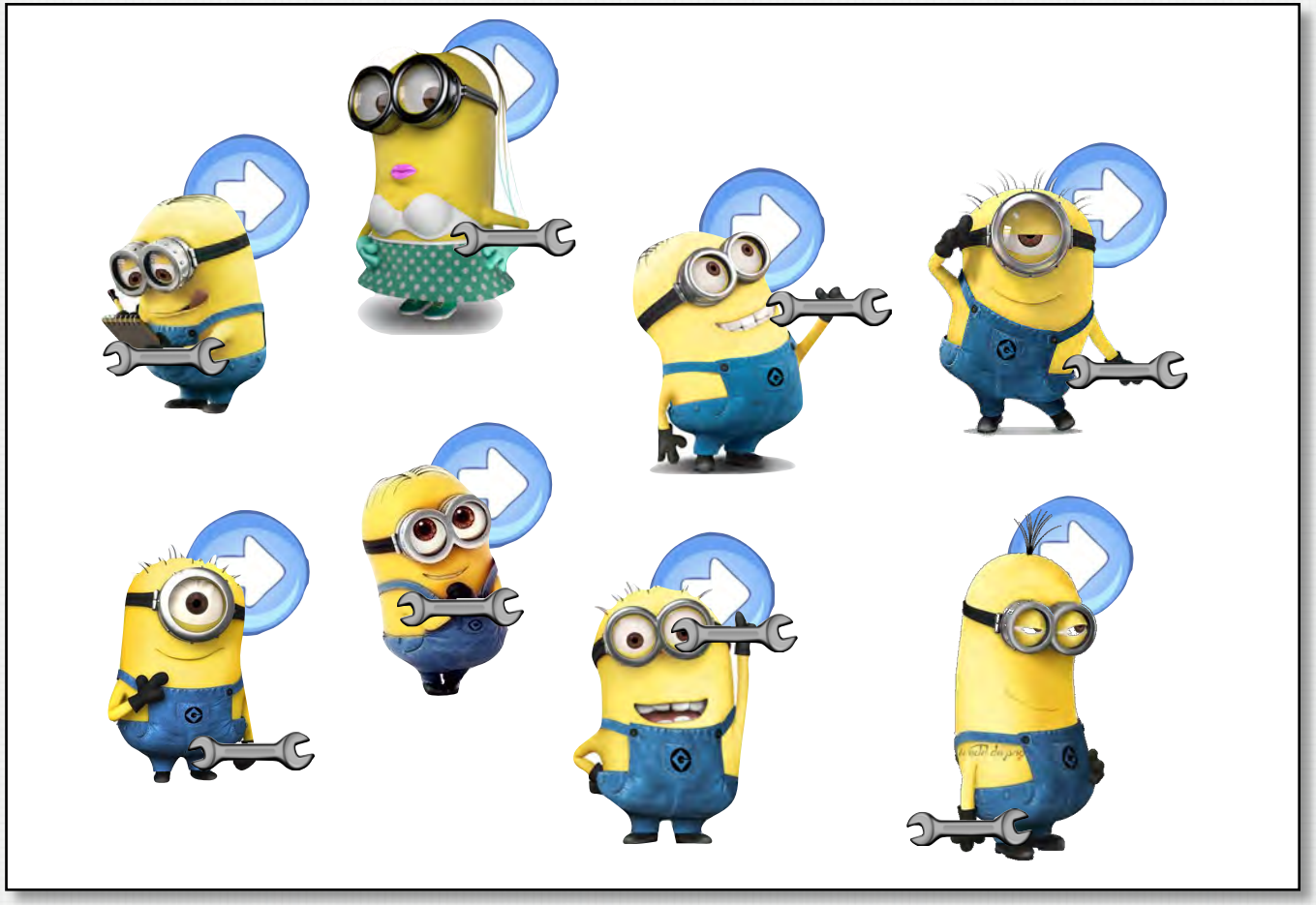
- Reaches across all software development phases
  - Each model uses the most appropriate formalism to express the model properties and composition rules relevant at current level of abstraction

Concern Characteristics (2)

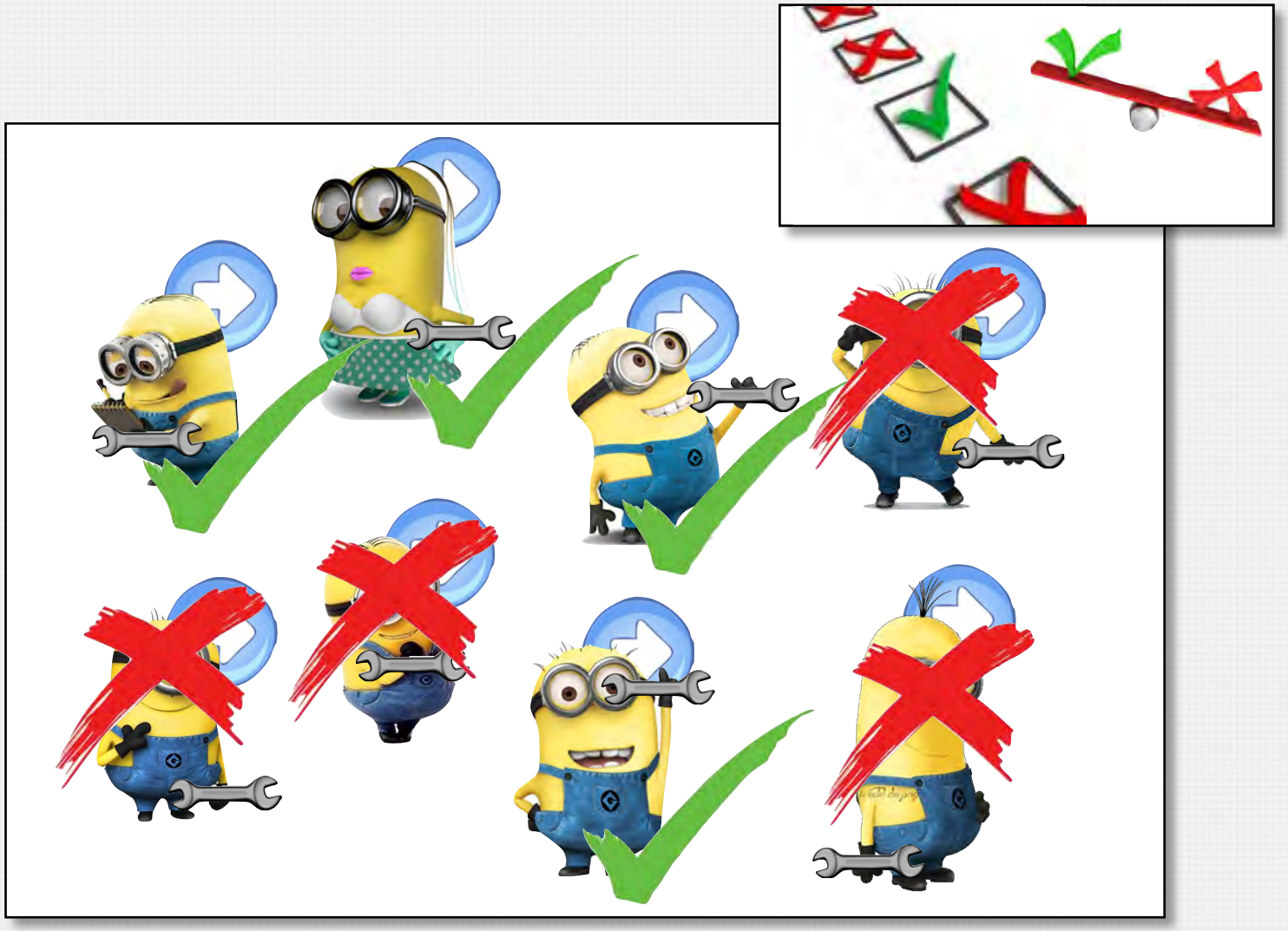
- Impact evaluation of variations
  - Modelled as generally as possible including all relevant variations
  - Guidance on how to choose among variations
  - Known impact on high-level system & stakeholder goals

- Defines model transformations
  - Link models/compositions from one phase to next
  - Avoid duplication of effort
  - Preserve properties

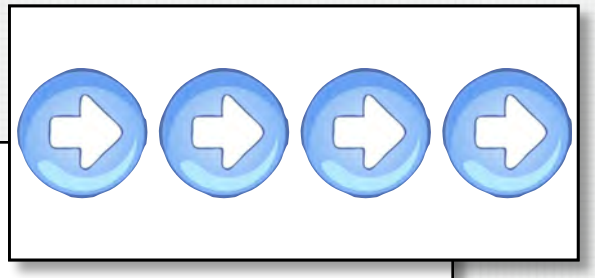
Example Concern

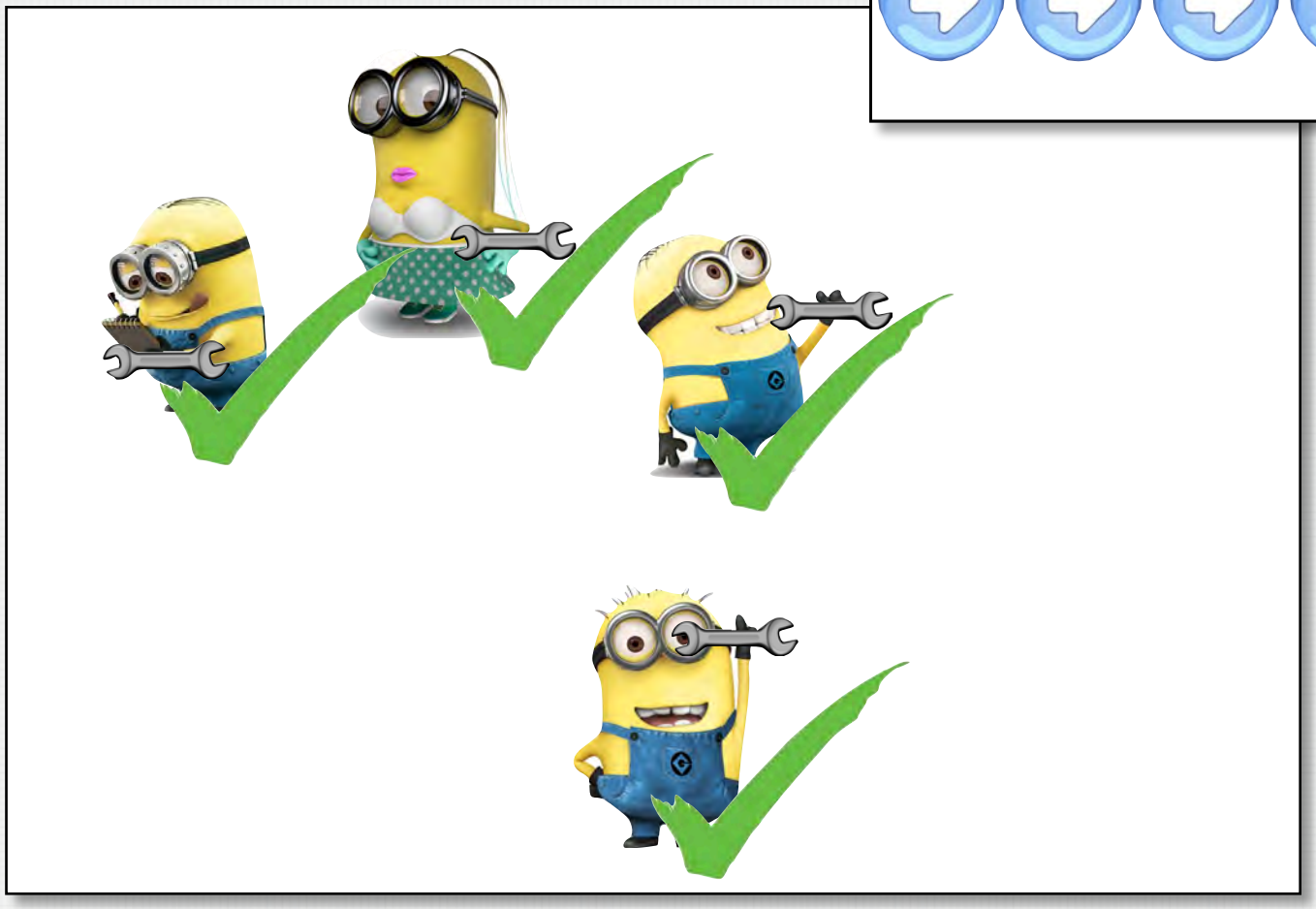


Variation Interface

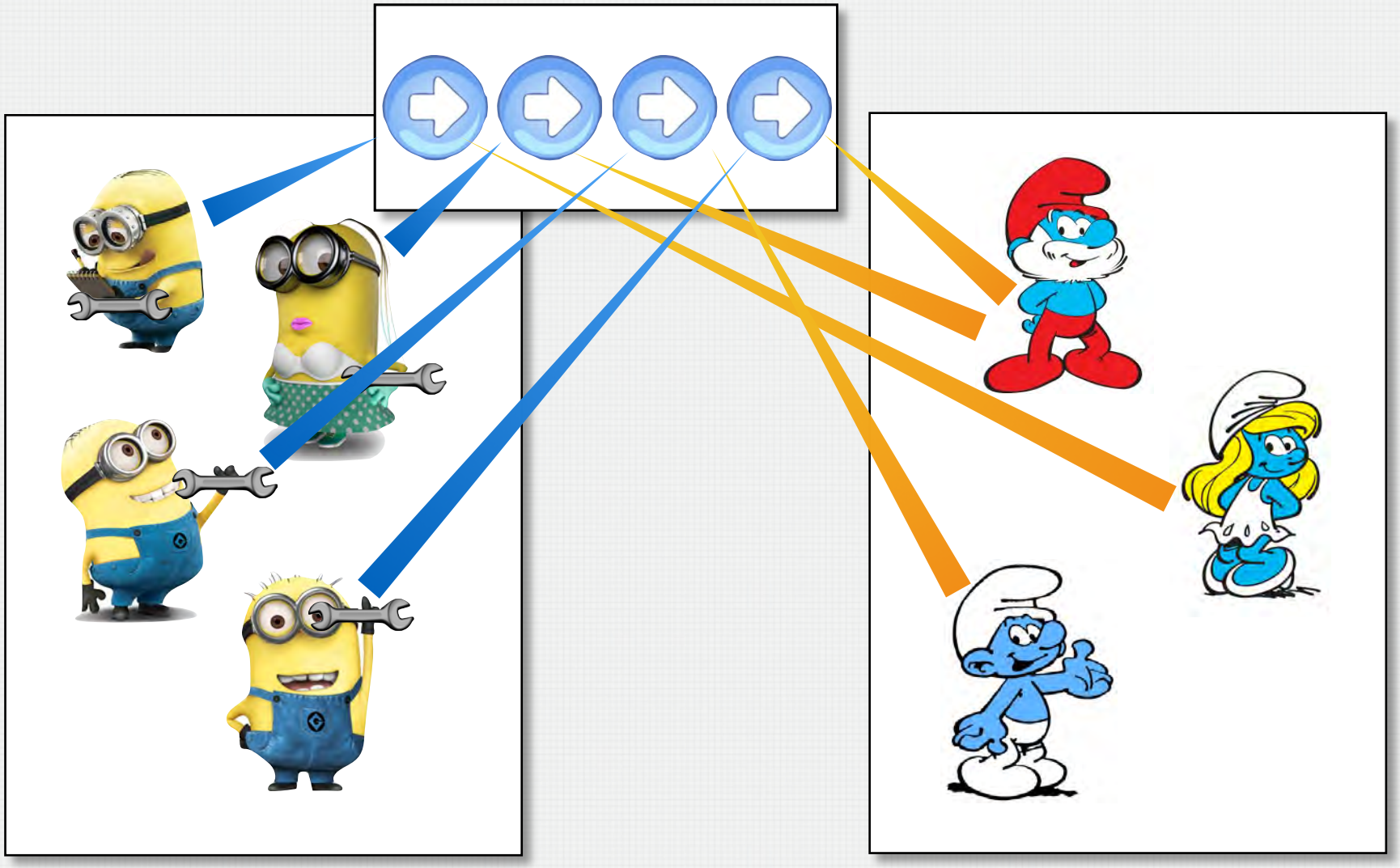


Customization Interface

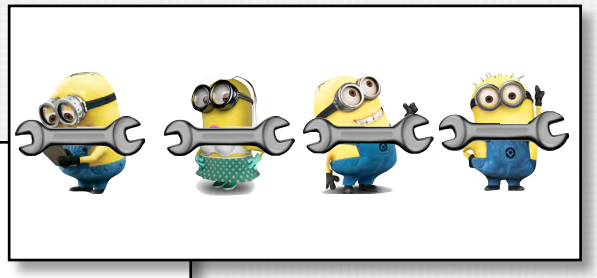


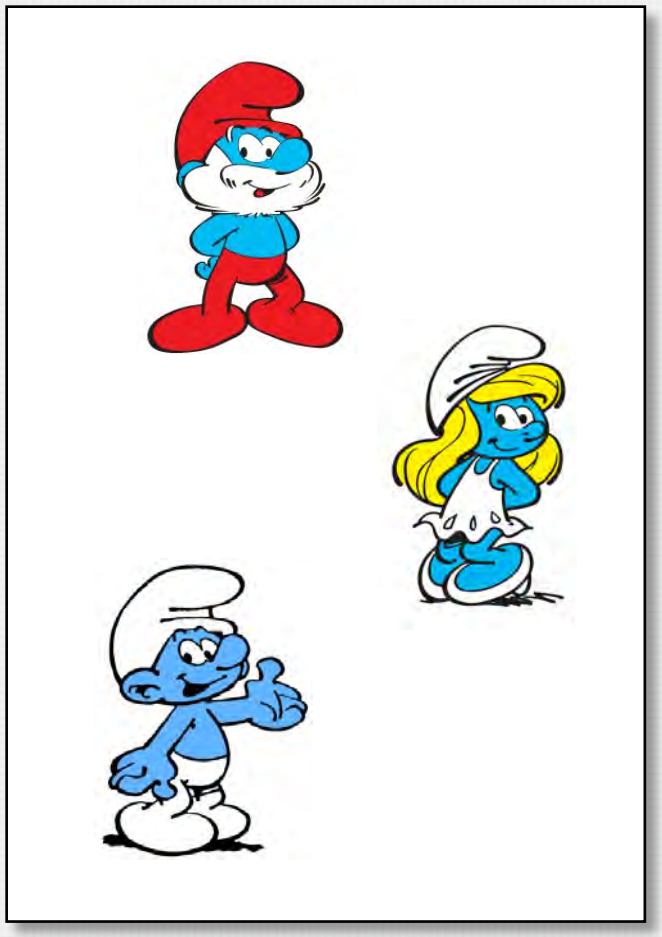


Customization Interface

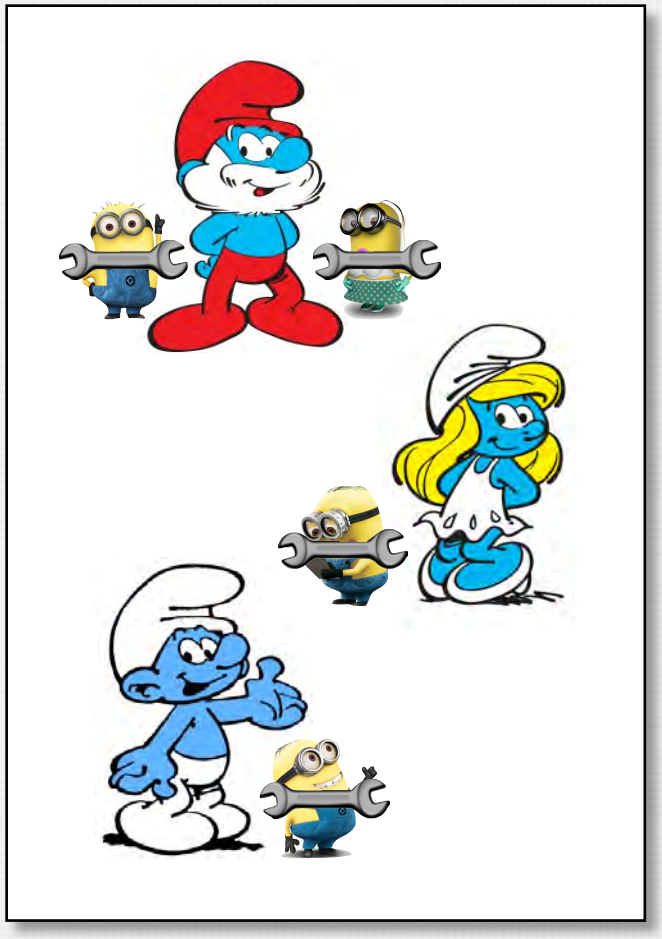


Usage Interface





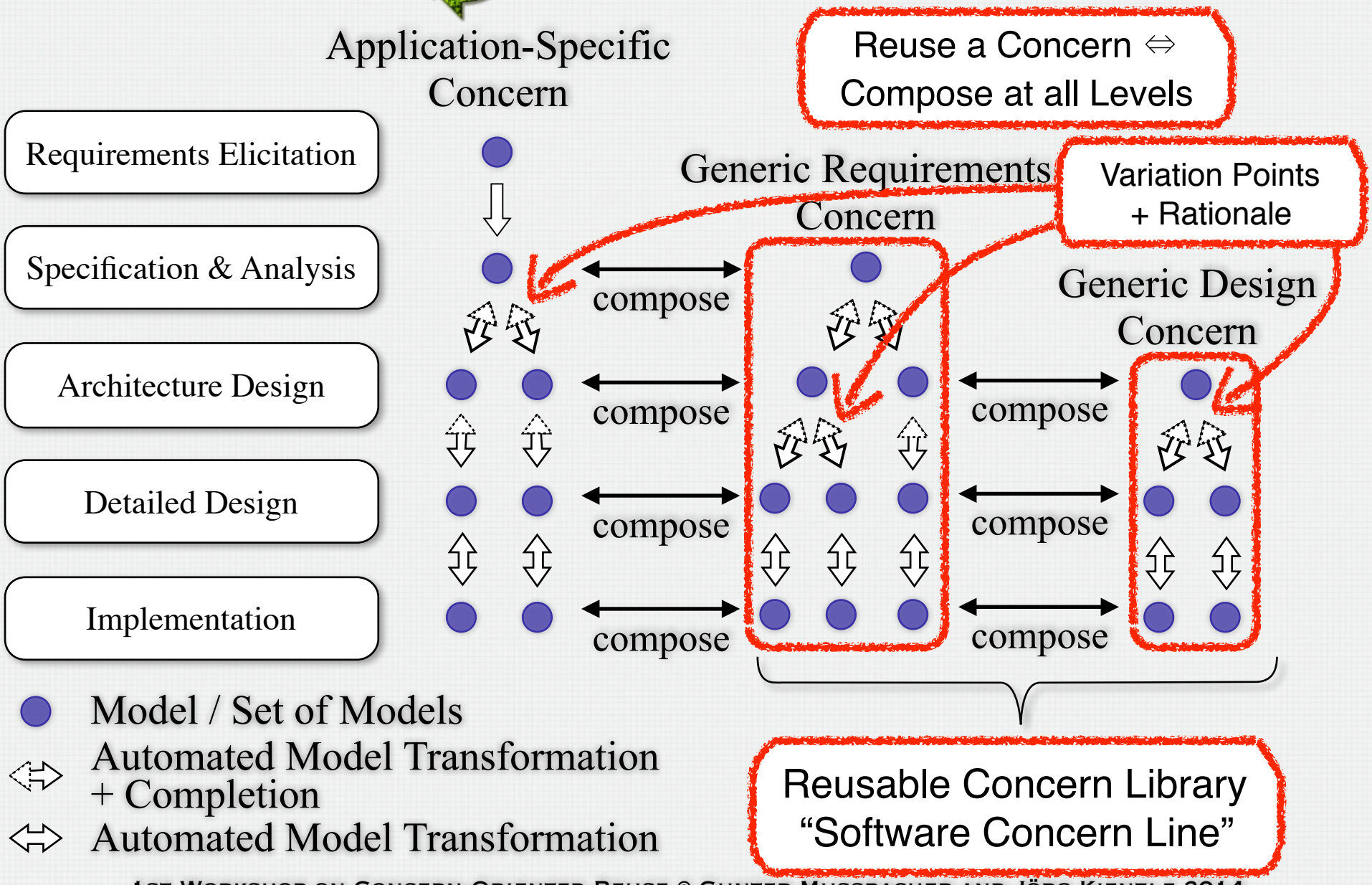
Composed Application



Concern Reuse Process

1. Use the variation interface of the concern to select the most appropriate feature(s)
   - That provides the desired functionality
   - That maximizes positive impact on relevant non-functional application properties
   - ➡ This generates the generic models for the selected feature(s) of the concern
2. Use the customization interface of the generated models to adapt the generic model elements to the application-specific model(s)
   - ➡ This generates the application-specific models for the selected feature(s) of the concern
3. Use the selected concern features within the application model(s) according to the usage interface

The CORE Approach



The CORE Approach