

# 25. Trustworthy Framework Instantiation

1

Prof. Dr. Uwe Aßmann  
TU Dresden  
Institut für Software- und  
Multimediatechnik  
Lehrstuhl Softwaretechnologie  
13-1.0, 02.01.14

- 1) The framework instantiation problem
- 2) Remedies



# Obligatory Literature

2

- ▶ Uwe Aßmann, Andreas Bartho, Falk Hartmann, Ilie Savga, Barbara Wittek. Trustworthy Instantiation of Frameworks. In *Trustworthy Components*, Reussner, Ralf and Szyperski, Clemens (ed.), Jan. 2006. LNCS 3938, Springer. Available at <http://www.springerlink.com/index/104074p5h8581115.pdf>

# 24.1 The Framework Instantiation Problem

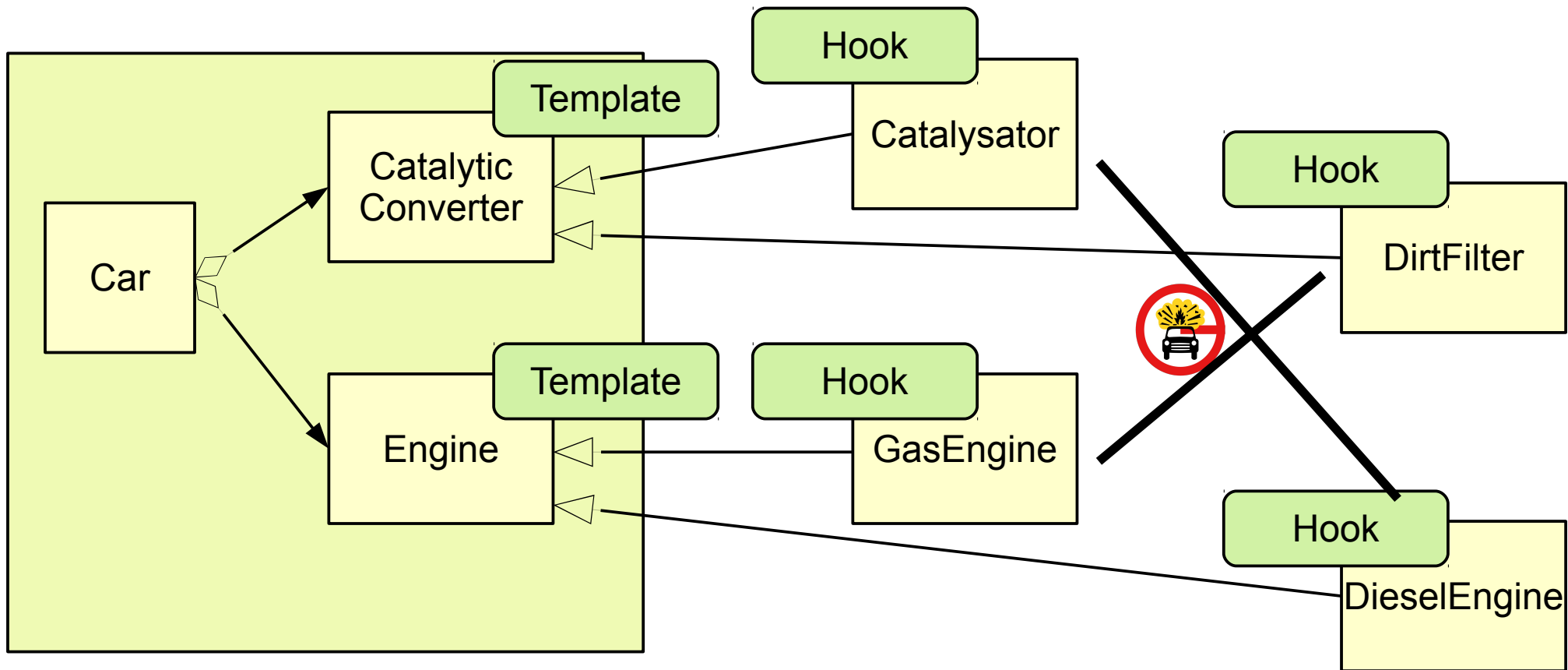
3

- ▶ Frameworks are often hard to instantiate
- ▶ Framework instantiation relies on **framework contracts**
  - ensuring typing on plugins
  - Whitebox frameworks are often instantiated with non-conformant subclasses
- ▶ Frameworks have many extension and variation points
  - and dependencies between them
  - Blackbox frameworks are often instantiated with non-fitting classes (*multi-point dependencies*)
- ▶ Some constraints cannot be checked statically, but must use dynamic contract checking

# Problem 1: A Car Configurator

4

- ▶ How to instantiate two 1-T-H hooks, if there are dependencies between them (*multi-point constraints*)?
- ▶ Static constraint, domain-specific



# Individual Configurators are a Big Business

5

► www.myboshi.net

The screenshot displays the myboshi.net website's beanie configurator. The browser address bar shows the URL [www.myboshi.net/Konfigurator/Itami.html](http://www.myboshi.net/Konfigurator/Itami.html). The page features a navigation menu with links for Konfigurator, Gutscheine, Story, Blog, Presse, Produktinfo, and Teamrider. A Facebook 'Gefällt mir' button shows 21 likes. The main content area is titled 'Design | Itami' and includes a 3D model of a green beanie with a colorful patterned band. To the right of the model are configuration options: 'Bommel' (checked 'nein', 'ja +5,00 €'), 'Größen' (checked 'M (52-56 cm)', 'L (57-60 cm)'), 'Erstfarbe:', 'Zweitfarbe:', and 'Drittfarbe:', each with a color selection palette. The price is listed as 40,00 € (including 19% MwSt. and shipping). The product description states a delivery time of 3-4 weeks and a composition of 30% Merino wool and 70% Acrylic. A 'Warenkorb' button is visible. The page also features a 'mykonto kasse warenkorb' menu and several certification logos on the right side.

# Individual Configurators are a Big Business

6

► www.shirtalarm.de

The screenshot displays the website www.shirtalarm.de, which is a T-shirt configurator. The main interface is divided into several sections:

- Navigation:** A top navigation bar includes links for Home, Grossauflagen, Kollektionen, Artikelübersicht, Motivgalerie, Druckverfahren, and Bestellinfo. There are also links for Login and Passwort vergessen?.
- Product Selection:** A central area shows a red t-shirt with the text "Drucken Sie hier **Motive Texte und eigene Fotos** auf Ihr Shirt!". Below the shirt, there are options for "Artikelgröße" (3XL, 4XL, 5XL) and "Artikelfarbe" (red, grey, dark blue, white).
- Text Options:** A panel on the right allows users to add text, motives, and photos to the t-shirt. It includes input fields and a "Flexdruck" option.
- Shopping Cart:** A section on the right shows "Ihr Warenkorb" with "0 Produkt(e) im Wert von 0,00 €". It includes a "rechnung" button and a "PRINTED IN GERMANY" badge.
- Large Orders:** A section titled "Grossauflagen ab 50 Stück" lists benefits such as "Angebot binnen 48h", "persönliche Fachberatung", "Top Qualität", "eigene Produktion direkt im Haus", "über 15 Jahre Erfahrung", "professionelle Auftragsabwicklung", "super Preise", "sehr kurze Lieferzeiten", and "riesen Auswahl an bedruckbaren Artikeln".
- Footer:** A red navigation bar at the bottom contains links for "Login", "FAQ Hilfe", "Gutscheine", "Mengenrabatt", "Partnerprogramm", "Kunden werben Kunden", "Newsletter", and "Infoseiten".

Shirtalarm bietet hochwertigen, günstigen T-Shirt Druck. Im T-Shirt Shop einfach mit Ihren Motiven Ihr persönliches T-Shirt bedrucken. Egal ob Kinder T-Shirts, lustige T-Shirt Motive, Funshirts oder T-Shirts mit Foto – wir drucken Ihr T-Shirt! Gestalten Sie selbst Ihr eigenes T-Shirt und heben Sie sich von der Masse ab. Auch als Geschenk oder zu besonderen Anlässen eignen sich bedruckte T-Shirts perfekt. Beim T-Shirt Druck steht Shirtalarm für höchste Qualität. Sie können Ihr T-Shirt drucken als Einzelstück oder in Großaufgabe – zu besonders günstigen Konditionen. Wir führen rund 100 verschiedene Artikel (T-Shirt, Tasse, Kissen, Jacken, Hoody...) und unzählige Motive für Ihren T-Shirt Druck. Falls Sie sich inspirieren lassen wollen, stöbern Sie doch mal durch unsere Kollektionen zum T-Shirt Druck. Hier bieten wir Ihnen alles von Fußball- über Tier- bis hin zu lustigen Kinder-Motiven. Wenn Sie Ihr individuelles T-Shirt bedrucken wollen und das zum günstigen Preis, dann sind Sie bei Shirtalarm also genau an der richtigen Adresse!

The bottom of the image shows a Windows taskbar with several open PDF files:

- Pongratz\_Tramm\_Wilb...pdf
- faulty\_epcs\_in\_the\_sa...pdf
- lamsweerde-goal-orie...pdf
- lamsweerde-04RE-rea...pdf

The system tray on the right contains the text "Alle einblenden".

# Individual Configurators are Frameworks

7

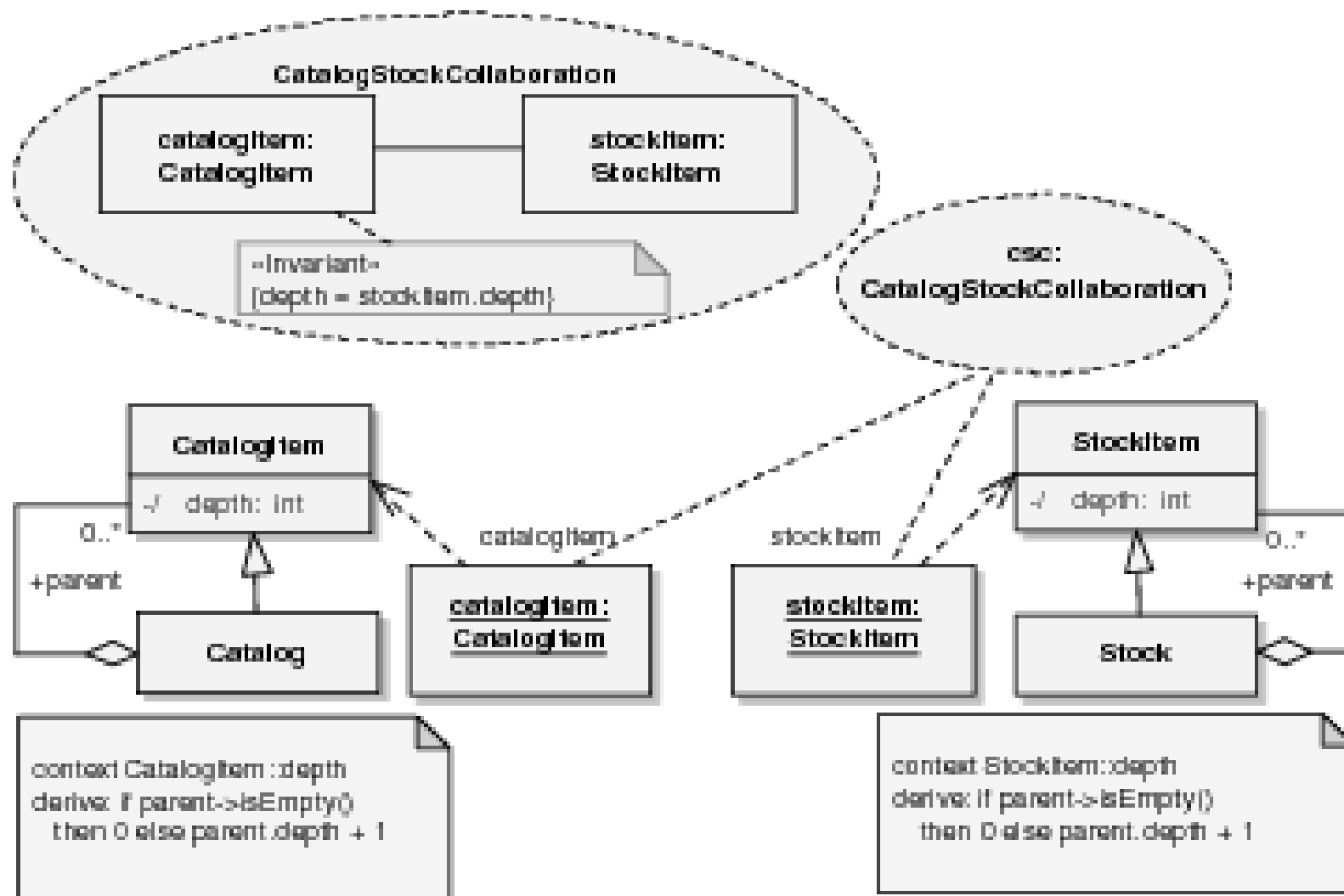
- ▶ Nowadays, you can buy the framework software for Individual Configurator Web Sites, e.g., <http://www.shirt-software.de/>
- ▶ The configurator frameworks must be adapted to a domain (which domain is not yet covered?) and to a company (individualization)

The screenshot shows the website for 'Shirt-Software' by 'tuffi Internet Vertriebs GmbH'. The page features a navigation menu with links like 'Startseite', 'Bestseller', 'Registrieren', 'Anmelden', 'Ihr Konto', 'Kontakt', and 'News'. A search bar is present with the text 'Suchbegriff eingeben'. The main content area is titled 'Katalog' and contains a promotional banner for 'NEUE VERSION' (New Version) of the 'Shirt-Software v2.0'. The banner includes a gold medal icon with 'Shirt-Software 2.0' and a product box image. Below the banner, there is a paragraph describing the interactive product designer and its capabilities. The footer of the page indicates 'Professional Edition (v1.3)' and 'Shirt-Software Version 2'. The browser's address bar shows the URL 'www.shirt-software.de/?gclid=CN7moxo\_4rQCFQhwzAodNnQA\_g'.

# Problem 2: SalesPoint Framework

8

- ▶ Catalog and Stock hierarchies must be isomorphic
- ▶ Dynamic constraint; domain-specific

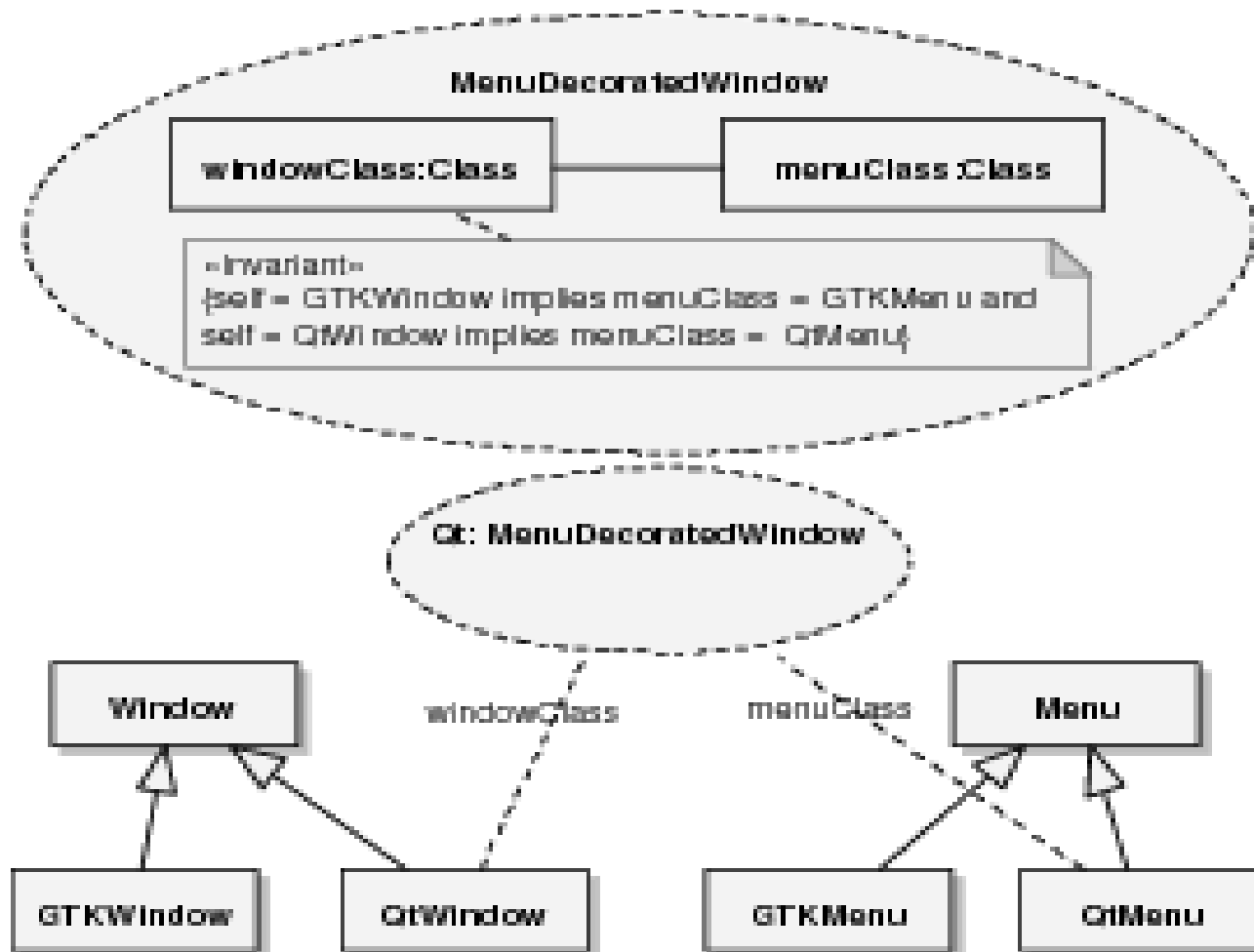




# Problem 3: Parallel Hierarchies

9

- ▶ Window types must be varied parallelly
- ▶ Static constraint, but technical



# Problem 4: Dynamic Assumptions

10

- ▶ Other dynamic contract checks

Null-checks  
Range checks  
Sortedness of ordered collections

Dynamic technical constraints

# Classification of Instantiation Constraints

11

|                |                                       | Facet 1: Stage                             |  |
|----------------|---------------------------------------|--|--|
|                |                                       | Static                                     | Dynamic  |
| Facet 2: Cause | Domain-specific<br>(analysis-related) | Car configurator<br>multi-point constraint | SalesPoint<br>isomorphic hierarchies<br>of Catalogs and Stocks |
|                | Technical<br>(design-related)         | Windows parallel<br>hierarchies            | Dynamic assumptions<br>Dynamic contracts                       |



# 24.2 Remedies for Trustworthy Instantiation

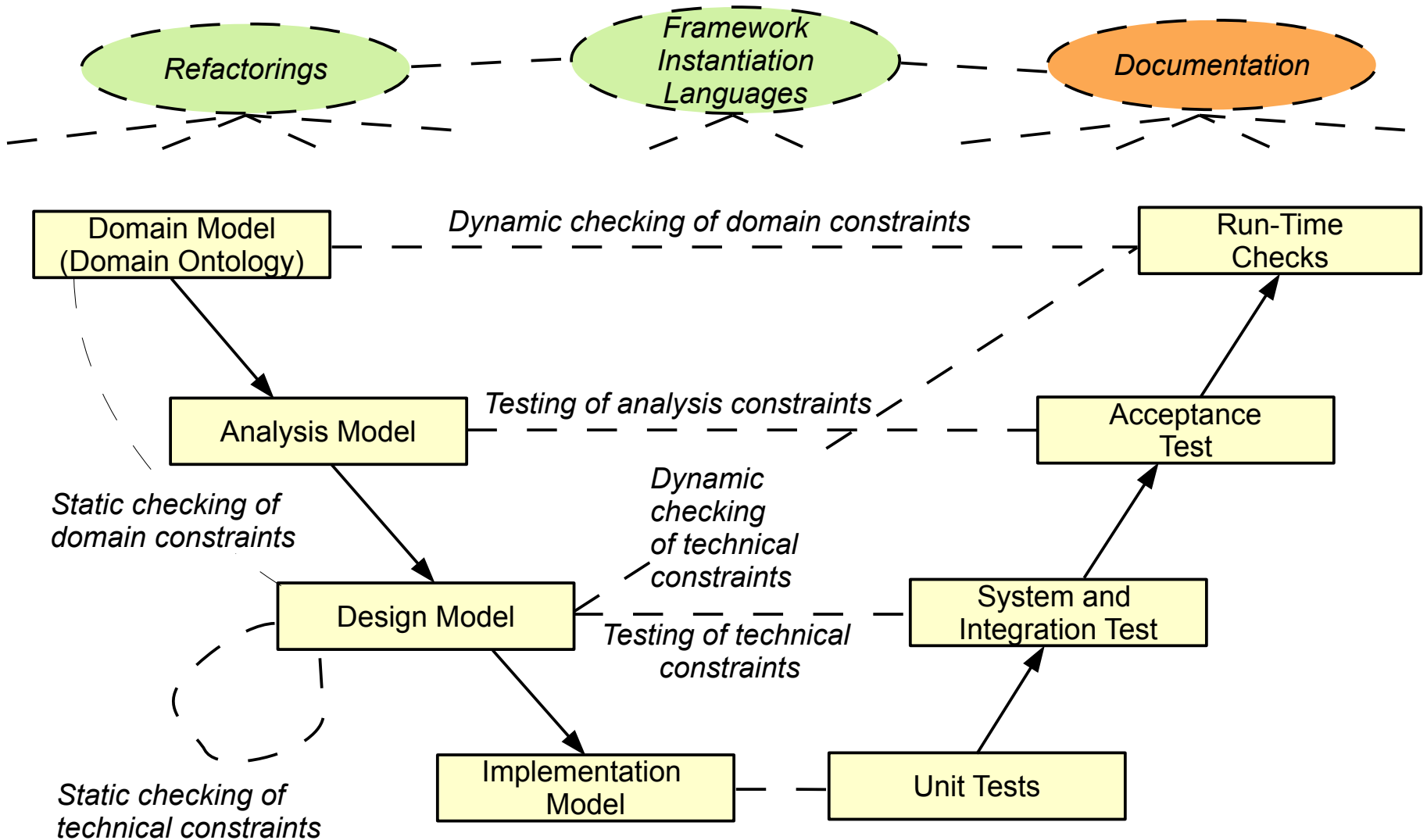
---

12



# Checking Mechanisms in All Phases of the Life Cycle

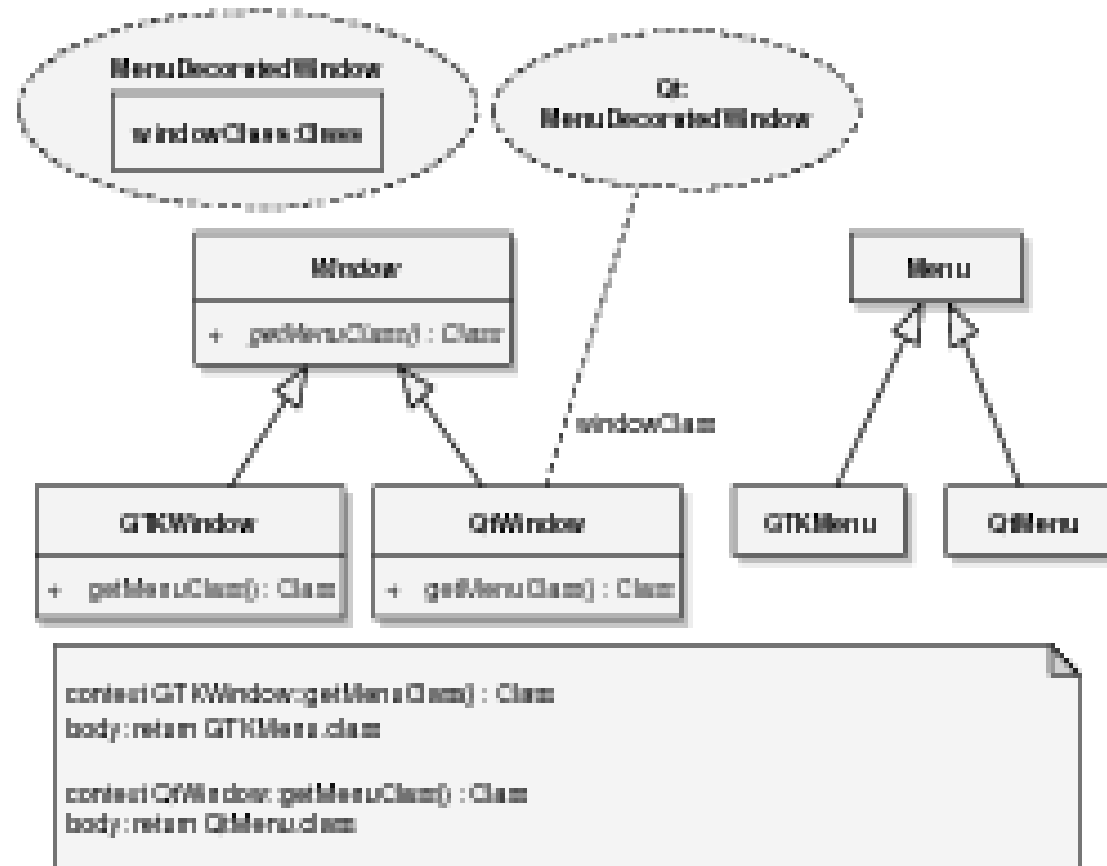
13



# Remedy 1: Refactoring of Multi-Point Constraints

14

- ▶ Multi-point constraints can be refactored such that the constraint moves inside the framework
  - One point is removed
- ▶ Advantage: Framework can control itself

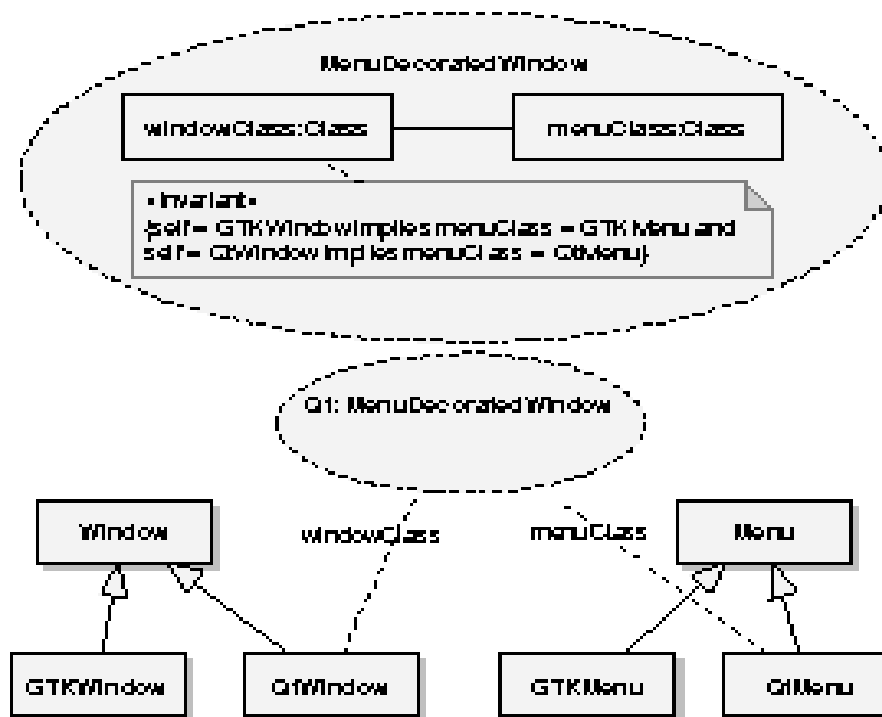


# Remedy 2:

## Static Verification of Static Constraints

15

- ▶ *UML collaborations* are appropriate to describe static (technical and domain-specific) instantiation constraints.
  - OCL specifies static invariants of the framework, instantiation preconditions and postconditions
  - OCL can reason over types, hence, instantiations or extensions of the framework can be analyzed and verified



# Remedy 3: Framework Testing

16

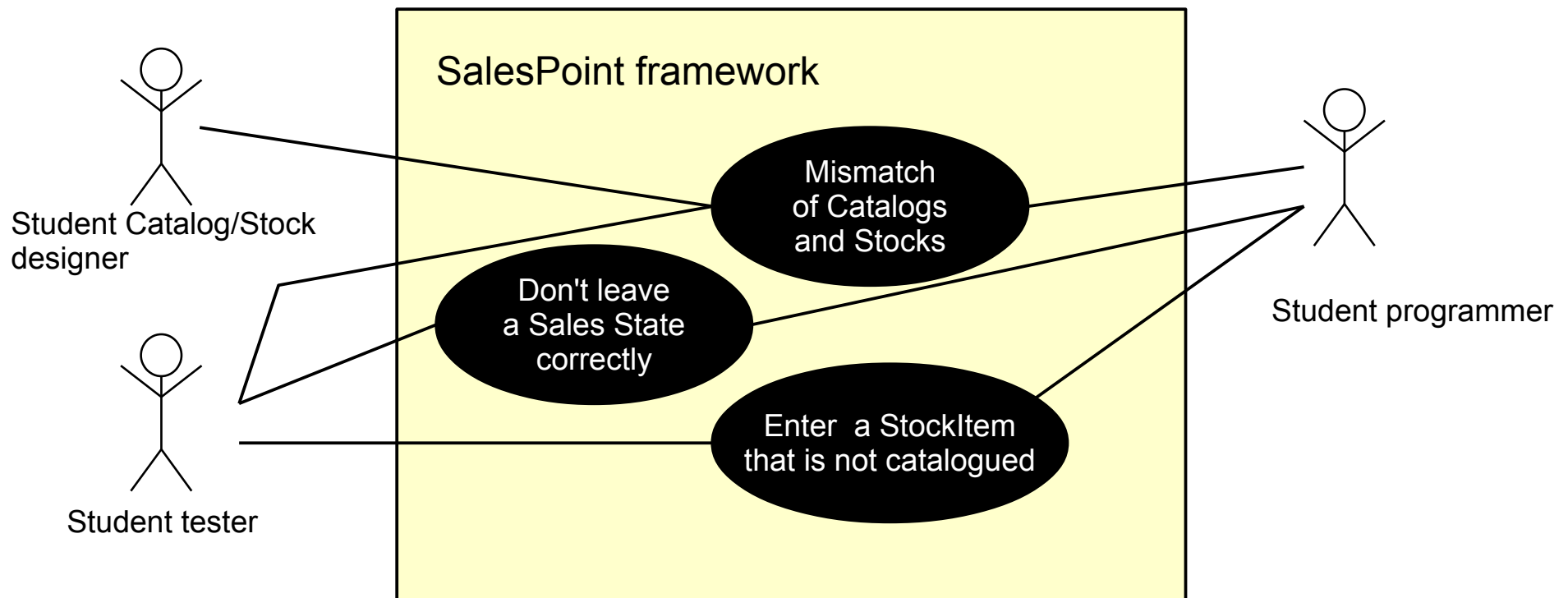
- ▶ Frameworks must be *negatively tested*
  - Beyond functional tests (positive tests), censorious negative tests for the behavior in case of misinstantiation must be conducted
  - Negative test cases have to be derived
    - specifying ill instantiation conditions
    - and the behavior of the framework
  - Framework must react reasonably
    - NOT dump core
    - Handle exceptions appropriately
    - Emit comprehensible error messages, also to the end user



# Misuse Diagrams

17

- ▶ *Misuse diagrams specify misuse cases*, dually to use case diagrams, which specify functional use cases
- ▶ [Sindre, G., Opdahl, A.L. Eliciting security requirements with misuse cases. Requirements Engineering 10 (2005) 34–44]
- ▶ Used to describe system abuse (intrusion, fraud, security attacks)
- ▶ Coarse-grain technique to specify also *framework misuse*



# Negative Test Table Entries

18

- ▶ From use case diagrams, usually test tables are derived
  - A test table contains test case entries, describing one test case
    - Class of test case (positive, negative)
    - Onput parameters of method
    - Output parameters
    - Reaction, state afterwards

| Testcase | Testclass | Input            | Output         | Reaction |
|----------|-----------|------------------|----------------|----------|
|          |           | String date      | Date d1        |          |
|          |           |                  | day month year |          |
| 1        | positive  | 1. Januar 2006   | 1 1 2006       |          |
| 2        | positive  | 05/12/2008       | 5 12 2008      |          |
| 3        | positive  | January 23, 2007 | 23 1 2007      |          |
| 4        | negative  | Mak 44, 2007     |                | failure  |
| 5        | negative  | March 44, 2007   |                | failure  |

# Negative Test Case Entries for Misuse of Frameworks

19

- ▶ Input parameters must be refined
  - Dynamic constraints are tested as usual negative test cases, with input and output parameter specification
  - Static constraints, however, work on types. Hence, their test case entries are different. Negative test cases specify ill instantiations, framework error messages and exception handling

| Testcase | Testclass   | Input   |           | Reaction                                      |  |
|----------|-------------|---------|-----------|---|--|
|          |             | hook 1  | hook 2    |   |  |
|          |             |         |           |   |  |
| 1        | pos. static | QtMenu  | QtButton  |   |  |
| 2        | pos. static | GtkMenu | GtkButton |   |  |
| 3        | neg. static | QtMenu  | GtkButton | error „for multi-point, use parallel classes“ |  |
| 4        | neg. static | GtkMenu | QtButton  | error „for multi-point, use parallel classes“ |  |

# Derivation of JUnit Test Cases

20

- ▶ From every test table entry dealing with a dynamic constraint, a JUnit test case is derived ([www.junit.org](http://www.junit.org))
  - Test method or test class with test method, deriving from class *TestCase*
- ▶ From every test table entry dealing with a static constraint, a compilation test suite case is derived
  - Stored in a database
  - Sold with the framework to the customer of the framework
  - Helps the customer to instantiate right
- ▶ See course Softwaretechnologie II, summer semester

# Remedy 4:

## Framework Instantiation Languages

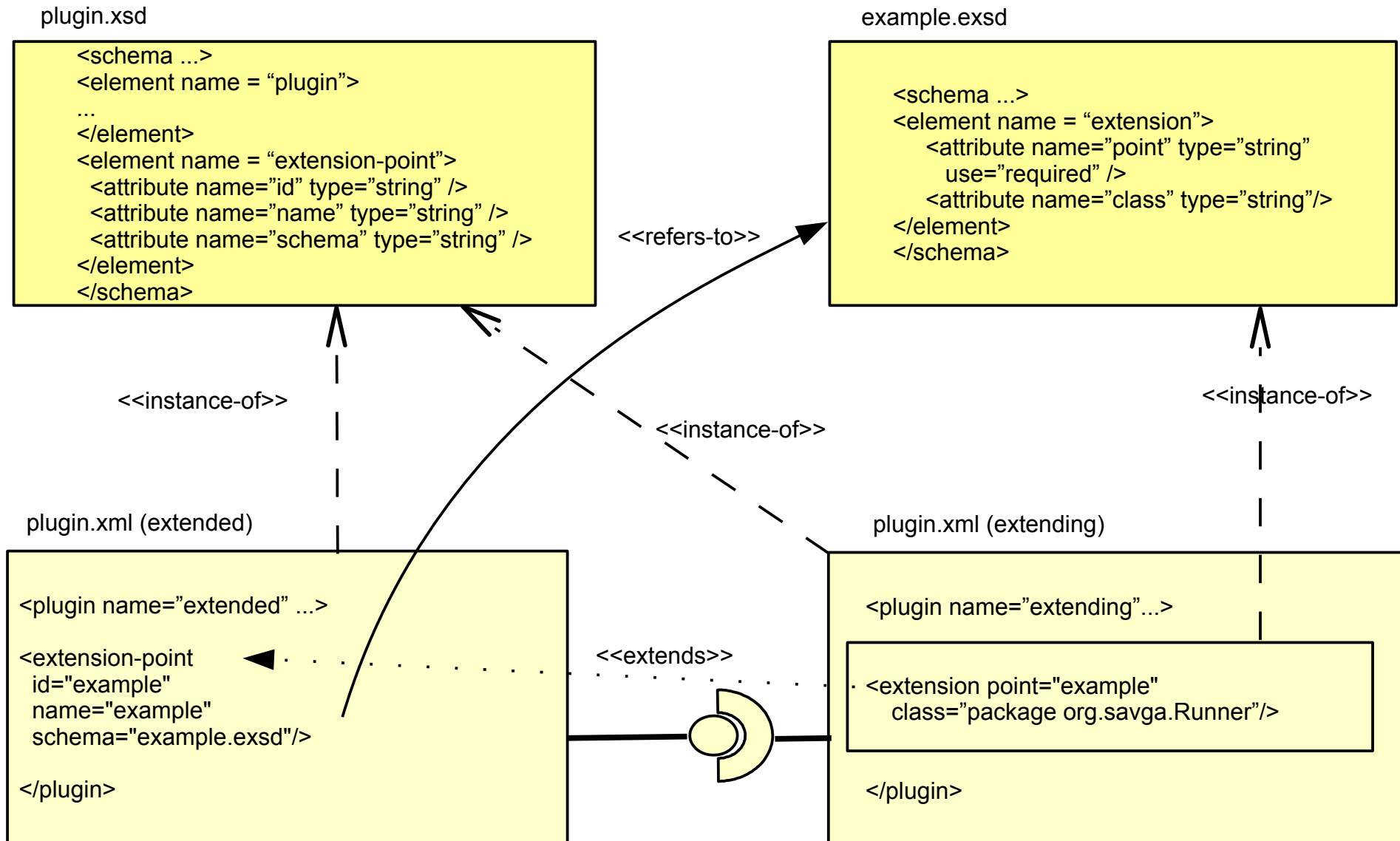
---

21

- ▶ Eclipse has demonstrated that a framework extension (instantiation) language can be beneficial
  - to type variability and extension points
  - to describe not only extension points for code, but also for other resources, such as GUI elements, business objects, etc.
- ▶ Eclipse language is based on XML, thus restricted on:
  - XML tree specifications
  - XML base types

# Eclipse Extension Specs

22



# Why A Framework Extension Language Should Be Based on Logic

23

- ▶ Beyond XML, logic can capture context-sensitive static constraints
  - also static multi-point framework instantiation constraints
- ▶ However, the logic must be enriched with domain-specific concepts, such as framework, hook, variation point, extension point, instantiation, etc.
- ▶ Good candidates are *typed logic languages*
  - Ontology languages OWL, SWRL
  - Frame logic (F-logic, on top of XSB)
  - OCL on UML class diagrams (UML collaborations)

# Remedy 5: Dynamic Contract Checking

24

- ▶ Dynamic multi-point constraints must be checked at run-time
  - Mainly, this amounts to *contract checking* of the framework
- ▶ Two best practices can be applied:
  - Framework contract layers
  - Contract aspects



# Framework Contract Layers

25

- ▶ Best practice is to check a dynamic constraint (single- or multi-point) in a separate layer, encapsulating the *contract concern*
- ▶ The checking layer is called from outside (the application), but the inner layer from inside the framework. This is much faster than checking always!
  - When composing the framework with others, the contract layer can be

```
class Collection {
    public boolean sorted() { ... /* sortedness predicate */ }
    public Element searchBinary(ElementKey key) {
        // contract checking
        if(!sorted())
            sort();
        // calling the inner layer
        return searchBinaryInternal(key);
    }
    // inner layer
    protected Element searchBinaryInternal(ElementKey key) {
        .. binary search algorithm ...
    }
}
```

# Remedy 6: Contract Aspects

26

- ▶ Once encapsulated in a layer, contract checks can be moved into a *contract aspect*
  - Tools such as Aspect/J can weave the contract in
  - Here: methods of package *framework* that have a parameter of type *Menu* are checked on null value
- ▶ Advantage: the aspect can easily be exchanged
  - Reduces effort, in particular when the aspect is *crosscutting*

```
before(Menu m): call(* framework.*.(Menu) && args(m) {  
    if (m == null) {  
        throw new Exception ("Null Menu parameter passed when " +  
            thisJoinPoint.getThis() + " was called ");  
    }  
}
```

# What Have We Learned?

27

- ▶ Framework instantiation and extension is hard, because there are many constraints, both domain-specific and technical, to obey
- ▶ Multi-point constraints describe dependencies between two or several framework hooks
- ▶ Appropriate remedies against misinstantiations are:
  - Thorough documentation (well, of course with the pyramid principle)
  - Refactoring (removal) of multi-point constraints
  - Negative testing with misuse diagrams and negative test table entries
  - Using logic to verify static constraints
  - Use contract layers and contract aspects to facilitate checking of dynamic constraints

# The End



28