
Andreas Lux

26.01.2011



**Verknüpfung unterschiedlicher Modellsprachen (BPMN, UML, DSL) zur
Anforderungsanalyse**



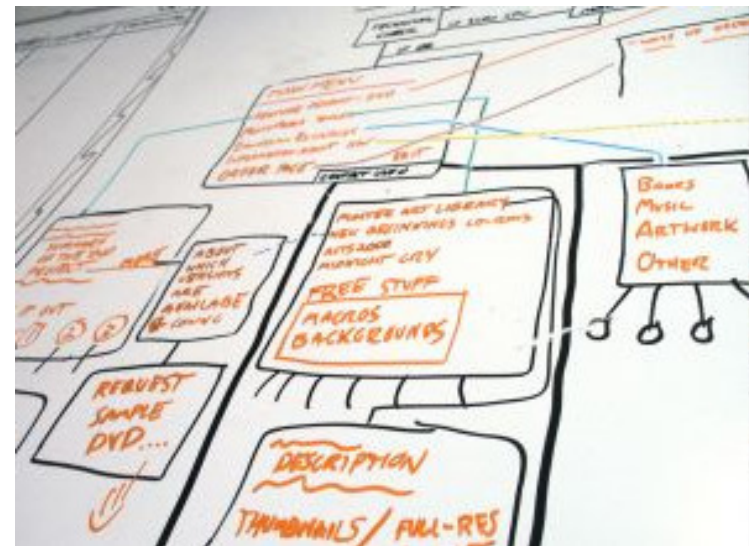
Warum unterschiedliche Sprachen?

Nicht alle Probleme eignen sich, um mit Standardsprachen beschrieben zu werden

Gründe können sein:

- zu umfassend
- nicht konkret genug
- nicht sprechen genug
- ineffizient in der Modellierung
- zu aufwändig bei der Integration von Generatoren

→ Die Lösung!?: **DSL**
(**D**omain **S**pecific **L**anguage)



Der Wunsch – Passgenaue Modellierungssprache

- Anforderungen an **DIE** Modellierungssprache:
- Zum Problem passender Sprachschatz
→ fachlich sprechende Notation
- Keine redundanten oder überflüssigen Sprachmittel
→ **eine** Lösung für **eine** Problemart
- Hohe Effizienz bei der Modellierung
→ Umschaltbare Notationen (textuell vs. grafisch)



Grafische vs. Textuelle Modellierung - was geht?

Grafische Modellierung

- Darstellung netzartiger Zusammenhänge (Flüsse, Strukturen)
- sinnvolle Darstellung von abstrakten Sichten



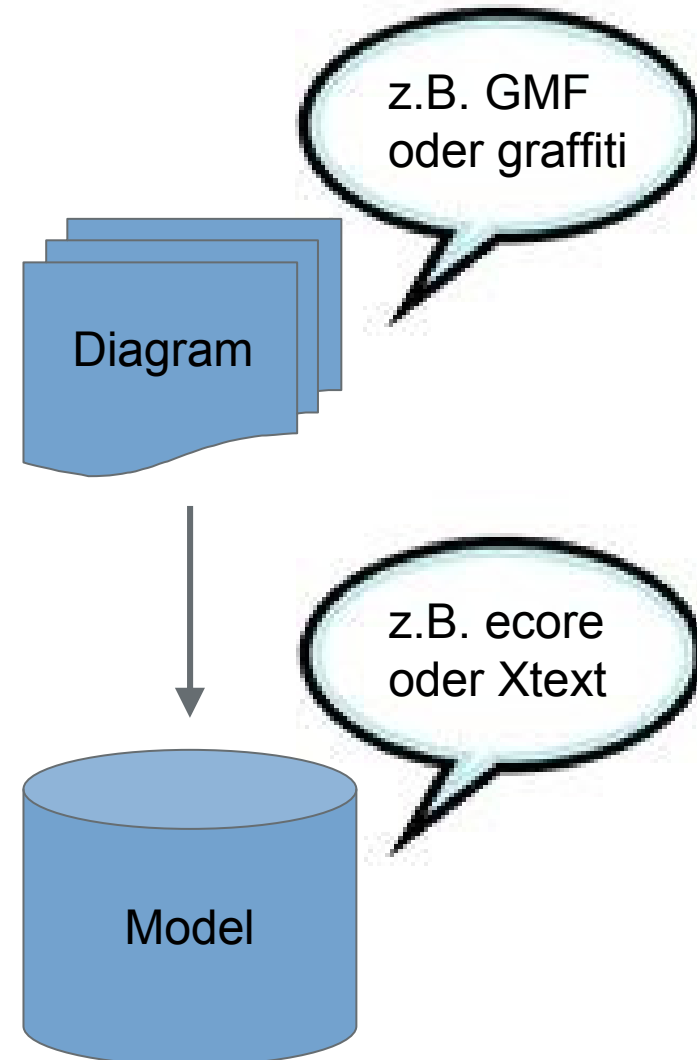
Textuelle Modellierung

- Sie aus wie Freitext, ist aber Modell!
- Klare Strukturen
- Einfache Sprache (wenige Regeln)
- Einfache Editierbarkeit

Umschalten zwischen Grafik und Text

Grundkonzept:

- Modell-Ressource enthält ausschließlich Modellinhalte
- Diagramme enthalten
 - | Referenzen auf Modellelemente
 - | Grafische Informationen
- Textuelle Modellierung
 - | Modifiziert ausschließlich Modelle
- Graphische Modellierung
 - | Modifiziert Modelle und Diagramme



→ Aktuell ist eine alternative Bearbeitung nur rel. aufwändig umzusetzen

Eigene DSL – was muss man tun?

- DSL ist in eclipse ein separates Plugin
- **Definition** der (fachlichen) Struktur
- **Generierung** des Tree-Editors über EMF
- Generierung des **eclipse Plugin Projekts**
- **Deployment** des Plugins über eine **Updatesite**



→ „prinzipiell“ verwendbar über den generierten Tree-Editor

Nachteil: relativ umständlich in der Handhabung

- Grafischer Editor
 - | Kann am Bsp. GMF ebenfalls aus DSL heraus generiert werden
 - | Generierung des eclipse Plugin Projekts
 - | Deployment des Plugins über eine Updatesite
 - | Alternative: graffiti

→ GMF muss noch um Features zum Handling erweitert werden

→ Graffiti ist noch sehr jung
- Textuelle DSL (am Beispiel xText)
 - | Definition einer eigenen Grammatik
 - | Generieren des.ecore Modells über xText Generator
 - | Deployment des Plugins über eine Updatesite

→ Sehr effizient in der Anwendung, kann aber u. U. sehr unübersichtlich sein

Welche Sprache für welchen Zweck?

Beispiele für unterschiedliche Sprachen:

- **UML**: sehr allgemeine und umfangreiche Modellierungssprache (in unterschiedlichen Ausprägungen!)
- **BPMN**: Modellierungssprache für Prozesse / Abläufe
- **SysML**: Systembeschreibungssprache basiert auf UML

- Von der OMG gibt es diverse weitere Entwürfe für Domänenspezifische Sprachen

- DSLs von eXXcellent solutions:
 - | orchideo|gui-prototype DSL
 - | orchideo|object DSL
 - | orchideo|aspect DSL
 - | orchideo|documents Templatesprache

Voraussetzungen für eigene DSLs

- Es wird ein modellgetriebener Ansatz im Projekt verfolgt
- Ausdrucksmöglichkeiten bestehender Sprachen entsprechen nicht dem Modellierungsproblem weil diese:
 - | zu viele Freiheitsgrade (= Fehler) zulässt
 - | es für ein Problem mehrere Modellierungswege gibt
 - | zu umständlich zu modellieren ist
 - | zu wenig exakt ist
 - | nicht verständlich genug ist (sich nicht an der Problem-Domäne orientiert)

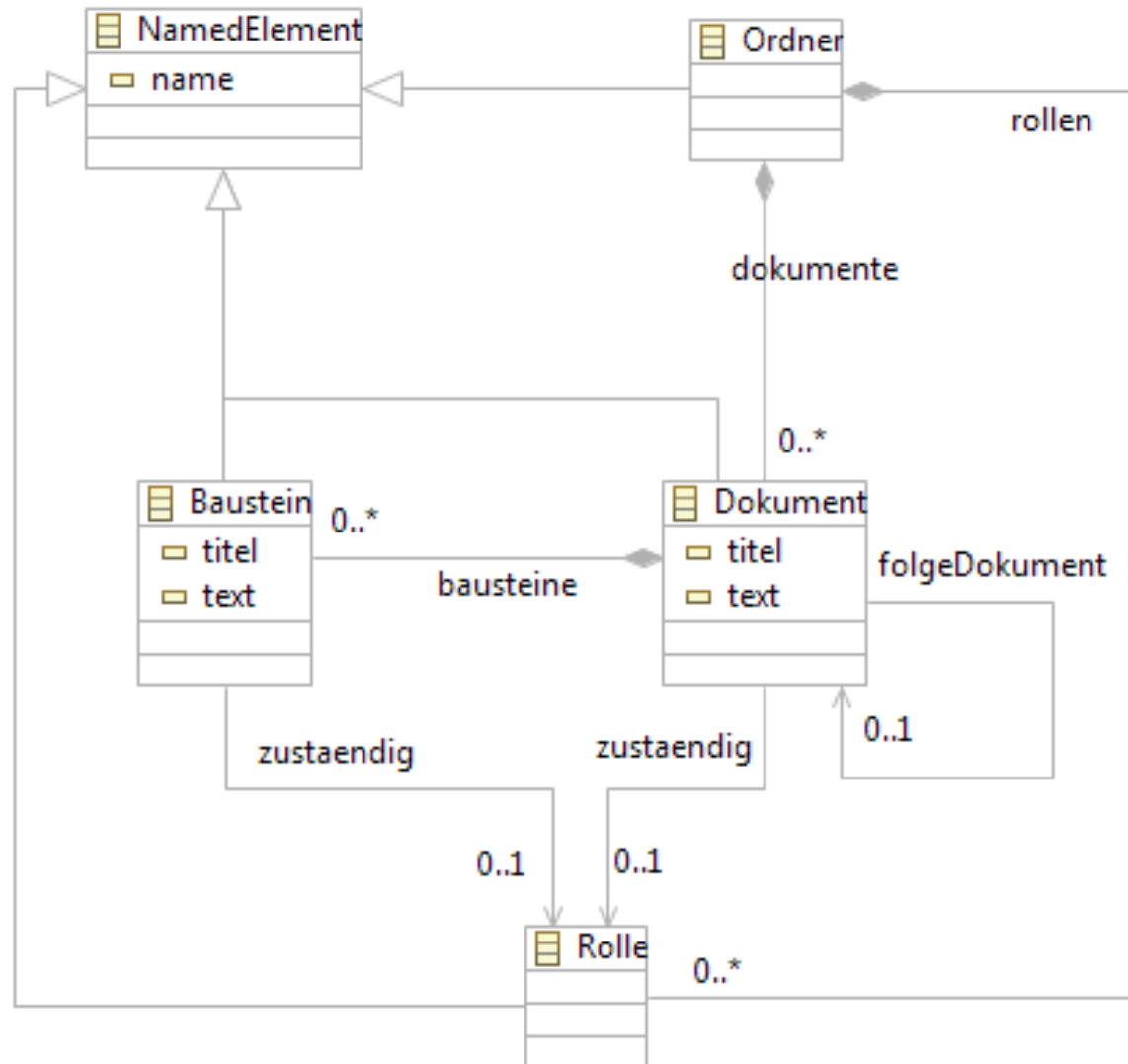


Beispiel: Dokumentendefinition

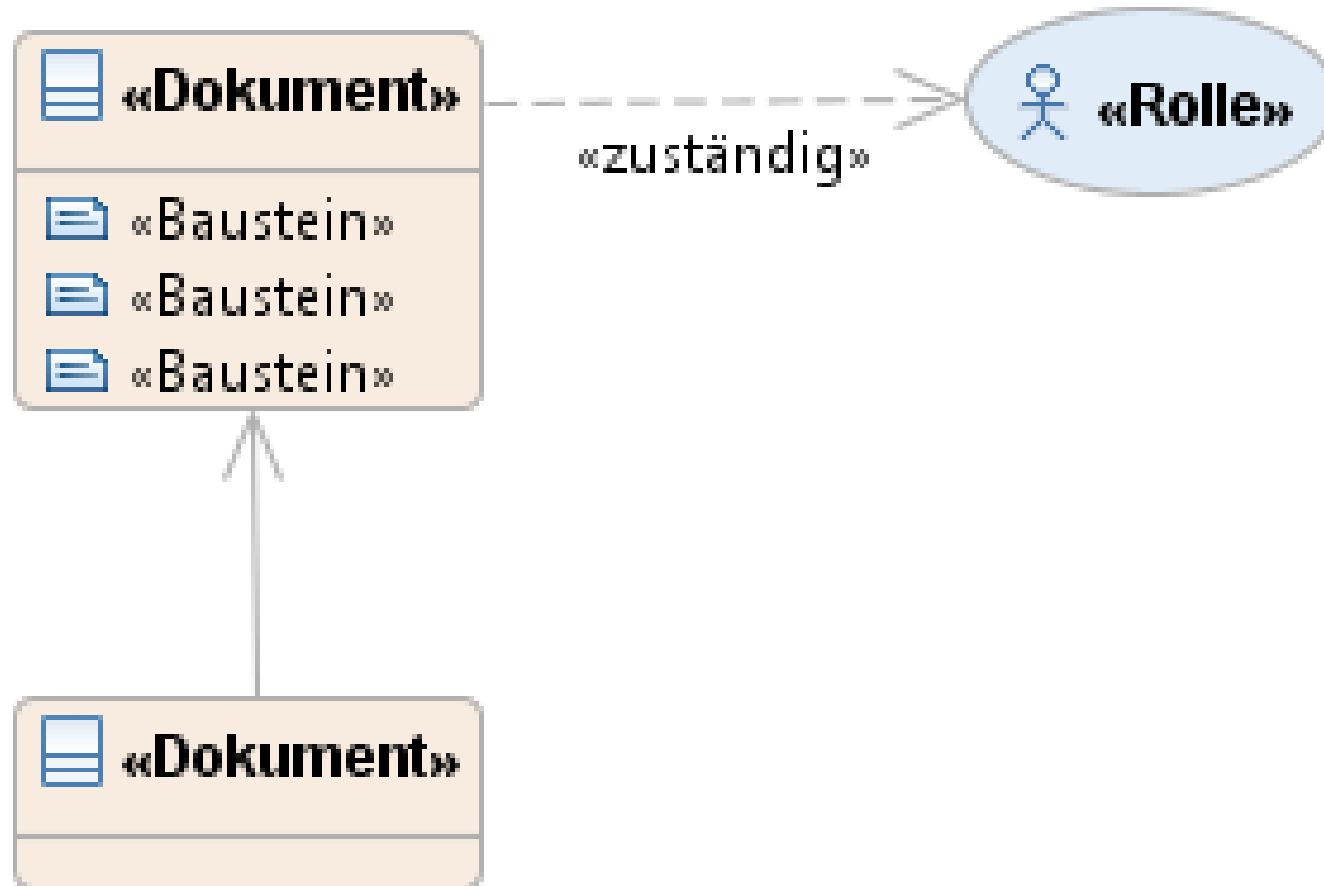
- Verwendung der Dokumentdefinition für eine Generator (nicht Teil der Präsentation)
- Es gibt unterschiedliche Dokumenttypen, die in unterschiedlichen Versionen vorliegen
- Dokumente bestehen aus Bausteinen
- Es gibt Zuständigkeiten pro Dokumenttyp und optional pro Baustein
- Die Versionshierarchie muss erkennbar sein
- Die Dokumenttypen soll grafisch modelliert
- Struktur und Inhalt der Dokumenttypen soll dokumentiert werden können



DSL-Definition

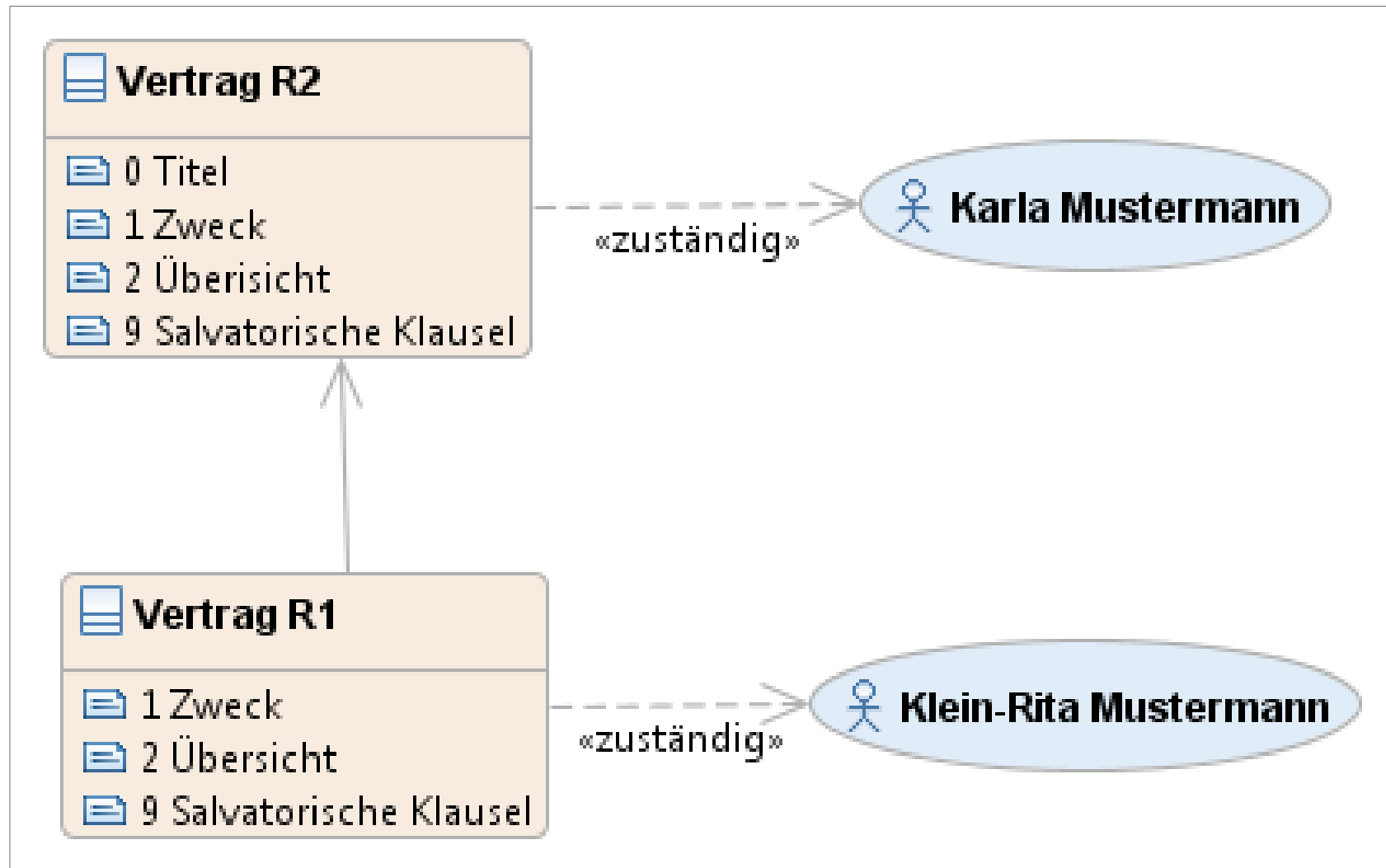


Wie sieht so was grafisch aus...

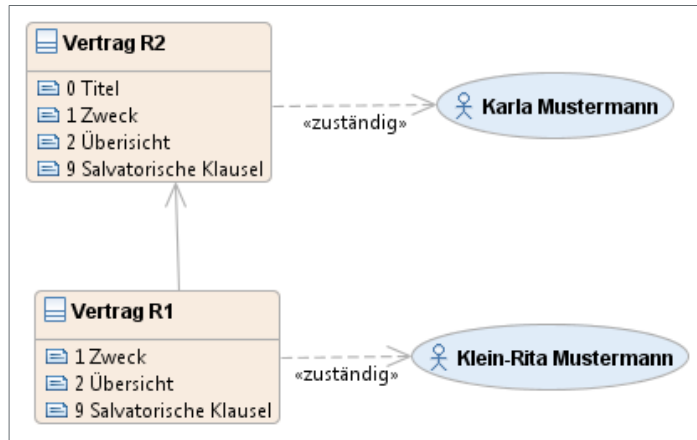


→ Beispiel 1: Neu anlegen live in der orchideo|suite

Und am konkreten Beispiel



Textuelle Repräsentation 1



definition for

```
"http://www.excellent.de/orchideo/  
Dokuments/1.0/DocManager.ecore"
```

```
new Dokument "Vertrag R2"
```

```
new Baustein "0 Titel"
```

```
new Baustein "1 Zweck"
```

```
new Baustein "2 Übersicht"
```

```
new Baustein "9 Salvatorische Klausel"
```

```
end Dokument
```

```
new Dokument "Vertrag R1"
```

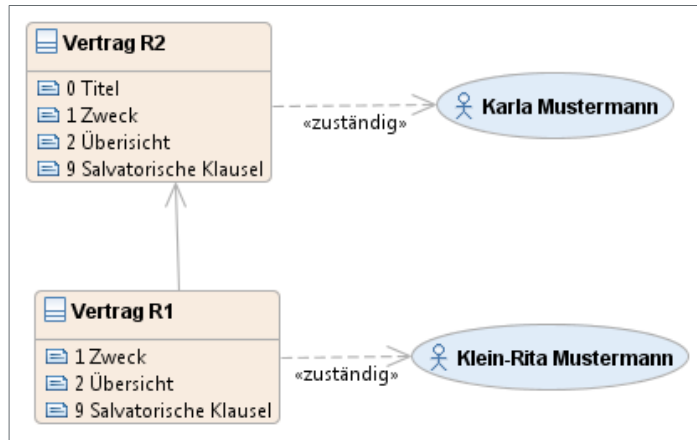
```
new Baustein "1 Zweck"
```

```
new Baustein "2 Übersicht"
```

```
new Baustein "9 Salvatorische Klausel"
```

```
end Dokument
```

Textuelle Repräsentation 2



```
new Rolle "Karla Mustermann"
end Rolle
```

```
new Rolle "Klein-Rita Mustermann"
end Rolle
```

```
new Dokument "Vertrag R2"
    Zuständigkeit "Karla Mustermann"
    new Baustein "0 Titel"
    new Baustein "1 Zweck"
    new Baustein "2 Übersicht"
    new Baustein "9 Salvatorische Klausel"
end Dokument
```

```
new Dokument "Vertrag R1"
    FolgeDokument "Vertrag R2"
    Zuständigkeit "Karla Mustermann"
    new Baustein "1 Zweck"
    new Baustein "2 Übersicht"
    new Baustein "9 Salvatorische Klausel"
end Dokument
```

Textuelle Repräsentation 3

```
new Dokument "Vertrag R1"  
  Titel: „Dies ist ein Titel zum aktuellen Vertrag“  
  FolgeDokument "Vertrag R2"  
  Zuständigkeit "Karla Mustermann"  
  new Baustein "1 Zweck"  
    Titel: „Dies ist ein Titel zum aktuellen Baustein“  
    Text: „Dies ist ein längerer Text Titel zum aktuellen Baustein.“  
  new Baustein "2 Übersicht"  
    Titel: „Dies ist ein Titel zum aktuellen Baustein“  
    Text: „Dies ist ein längerer Text Titel zum aktuellen Baustein. D  
    Titel: „Dies ist ein Titel zum aktuellen Baustein“  
    Zuständigkeit "Karla Mustermann"  
  new Baustein "9 Salvatorische Klausel"  
    Titel: „Dies ist ein Titel zum aktuellen Baustein“  
    Text: „Dies ist ein längerer Text Titel zum aktuellen Baustein. Dies ist ein  
    längerer Text Titel zum aktuellen Baustein. Dies ist ein längerer Text Titel zum  
    aktuellen Baustein. Dies ist ein längerer Text Titel zum aktuellen Baustein. Dies  
    ist ein längerer Text Titel zum aktuellen Baustein. Dies ist ein längerer Text  
    Titel zum aktuellen Baustein.“  
    Titel: „Dies ist ein Titel zum aktuellen Baustein“  
    Zuständigkeit "Karla Mustermann"  
end Dokument
```

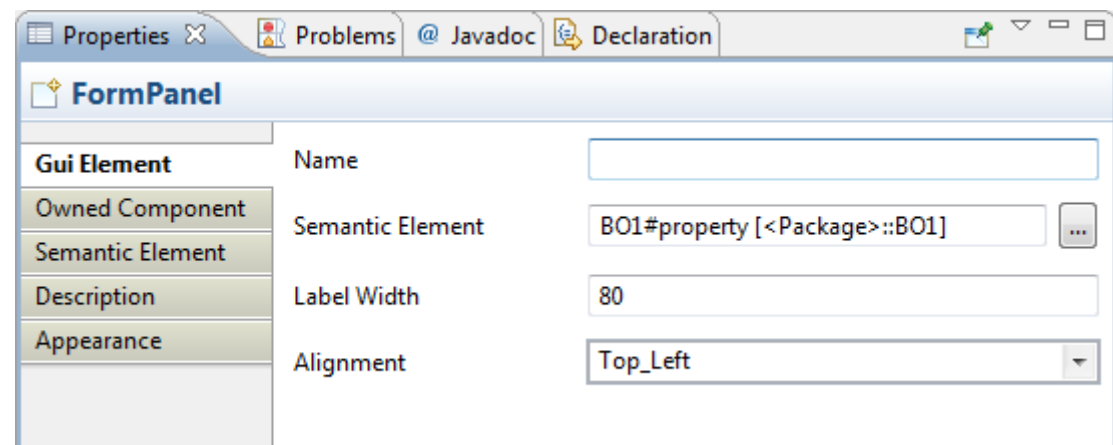
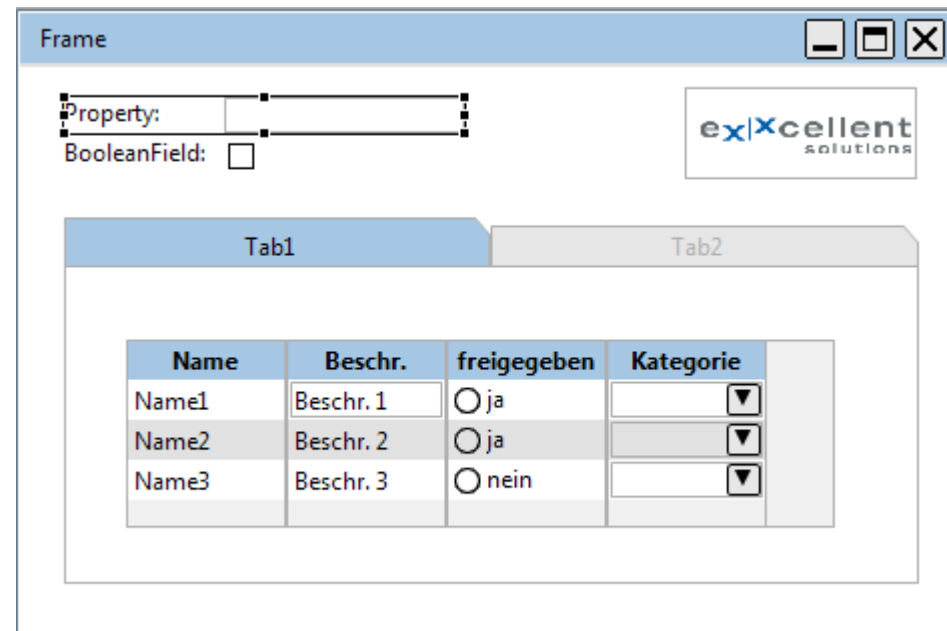
Beispiel 2: GUI-Modellierung - grafisch

Inhalte des Diagramms:

- Aufbau des Dialogs
- Metainformationen (z.B. zur Doku-Generierung, ...)

Inhalte der einzelnen Controls:

- Fachliche Attribute (Ref. zum BO-Modell, Beschreibungstexte, Regeln, ...)
- Technische Attribute (z.B. Größen, Reihenfolgen, Positionen, ...)
- Weitere Metainformationen



Beispiel 2: GUI-Modellierung – textuell 1

```
new Frame „Frame“
  new TextField „Property“ RelatesTo
  new BooleanField „Property“ Relates
  new TabPanel
    new Tab „Tab1“
      new Table
        new Column „Name“ Relates
        new Column „Beschr.“
        new Column „freigegeben“
        new Column „Kategorie“
      end Table
    end Tab
    new Tab „Tab2“
      new Tree Controller „de.ex
    end Tree
  end Tab
end TabPanel
end Frame
```

Frame

Property:

BooleanField:

ex|Xcellent solutions

Tab1 Tab2

Name	Beschr.	freigegeben	Kategorie
Name1	Beschr. 1	<input type="radio"/> ja	<input type="text"/>
Name2	Beschr. 2	<input type="radio"/> ja	<input type="text"/>
Name3	Beschr. 3	<input type="radio"/> nein	<input type="text"/>

Beispiel 2: GUI-Modellierung – textuell 2

```
new Frame „Frame“
  new TextField „Property“ RelatesTo
  new BooleanField „Property“ Relates
  new TabPanel
    new Tab „Tab1“
      new Table
        new Column „Name“ Relates
        end Column
        new Column „Description“
          RelatesTo „BO2#Descript
          Lable „Beschreibung“
          Alignment Left
          With „80“
          Description „Dies ist e
        end Column
        new Column ...
        ...
      end Table
    end Tab
  end TabPanel
end Frame
```

The screenshot shows a window titled "Frame" with a standard Windows-style title bar (minimize, maximize, close buttons). Inside the window, there is a form with the following elements:

- A label "Property:" followed by a text input field.
- A label "BooleanField:" followed by a checkbox.
- A logo for "ex|Xcellent solutions" in the top right corner.
- A tabbed interface with two tabs: "Tab1" (active) and "Tab2".
- Under "Tab1", there is a table with the following structure:

Name	Beschr.	freigegeben	Kategorie
Name1	Beschr. 1	<input type="radio"/> ja	<input type="text" value="v"/>
Name2	Beschr. 2	<input type="radio"/> ja	<input type="text" value="v"/>
Name3	Beschr. 3	<input type="radio"/> nein	<input type="text" value="v"/>

Rahmenbedingungen zum Einsatz von DSLs



Werden spezielle DSLs im Projekt eingesetzt, sollten folgende Rahmenbedingungen eingehalten werden können:

- Verwendung unterschiedlicher DSL innerhalb einer Umgebung
- Erweitern von „fremden“ DSLs
- Verknüpfen Modellinhalten aus unterschiedlichen DSL
- Generieren von Dokumenten



Verwendung unterschiedlicher DSL innerhalb einer Umgebung

Warum ist das wichtig?

- DSL dienen zur Beschreiben **eines konkreten Problemtyps**
- Viele andere Problemtypen können mit bestehenden Standard-DSLs (z.B. UML, BPMN, ...) bereits ausreichend gut abgebildet werden
- Man will die Welt nicht neu „erfinden“
- Die neue DSLs müssen sich strukturell mit bestehenden DSLs im Modellierungstool mischen und referenzieren lassen

→ Beispiel: UseCases und UI-Diagramme



Erweitern von „fremden“ DSLs

warum ist das wichtig?

- Erweitern bestehender DSL um projektspezifische Informationen (z.B.: „Description“-Feld bei UML2Tools UseCases, ...)
- Erweitern von (technischen) DSL um fachliche Aspekte (z.B.: mitführen einer UseCase ID, Tracker ID, Freigabeinformationen, ...)
- Erweitern von DSL um organisatorische Aspekte (z.B.: Referenzen auf weitere Inhalte)

→ Beispiel: UML2 Use Cases



Verknüpfen Modellinhalten aus unterschiedlichen DSL

Warum ist das wichtig?

- Systembeschreibungen sind sehr oft hierarchisch aufgebaut
- Sprachelemente verfeinern sich oft in weiteren Sprachelementen (z.B. UseCase → Activity → Sequence, UseCase → UI Diagramm)
- Sprachelemente referenzieren oft bestehende Definitionen (z.B.: Akteur → Klasse, UseCase → Service-Interface, „Dokument“ → Klasse oder Klassendiagramm, ...)

→ Beispiel: Use Case Overview



Generieren von Dokumenten aus gemischten Modelle für die Dokumentation / Kommunikation

Warum ist das wichtig?

- | Es gibt sehr oft Beteiligte (Kunde?), die keinen Zugriff auf das Modellierungswerkzeug haben
- | Erzeugung von Snapshots, falls das Modellierungswerkzeug keine „echte“ Versionierung beherrscht
- | Reduktion der Komplexität durch spezialisierte Dokumente (ausblenden der für den Leser unwichtigen / nicht relevanten Informationen)
- | Reuse von Modellinformationen für generierte Dokumente zur Vollständigen Dokumentation von Einzelteilen

→ **Beispiel: Dokument „Use Cases“**

Wenn das dann alles funktioniert...

Was sind die Vorteile bei Einsatz von spezialisierten DSLs?

- Allgemein:
 - | Erhöhung der Effizienz
 - | Wiederverwertbare Fachdefinitionen (auswertbar, generierbar, ...)
 - | Exakte Beschreibung der Fachlichkeit
- Grafische Modellierung:
 - | Guter Überblick
 - | Gute Abstraktion
- Textuelle Modellierung:
 - | sehr effizienter Editor



Wenn das dann alles funktioniert...

Was können die Nachteile sein?

- Zu hoher initialer Aufwand durch die Entwicklung der DSL und der zugehörigen Plugins
- Verlust der Allgemeingültigkeit der verwendeten Sprache (Kommunikation nach außen!)
- Zu hohe Aufwände bei der Wartung
- Vorgehen (noch) stark abhängig von der eingesetzten Toollandschaft



- Vorgehen ist dann sehr hilfreich, wenn die Modelle zur (Code-/Config) Generierung verwendet werden
- Die Einarbeitungsaufwände, um grafische Editoren „handhabbar“ zu machen, können sehr groß werden
- Einsatz von eigenen DSL für uns nur dann sinnvoll, wenn
 - | das Problem überhaupt abstrahiert werden kann
 - | Die Modelle (mehrfach) wiederverwendet werden können (z.B. Code **und** Doku)
 - | die Modellinformationen extern kommuniziert werden können (via Tool oder Doku)



Bleibt nur noch ein...

Danke!

... und herzlich willkommen an unserem Stand

<http://www.exxcellent.de/>

