Generische Implementierung

- Definition
- Konfiguration
- Begrifflichkeiten
- Aspekte

Definition

Menüaufruf: (Konfiguration) - (Regelwerke) - (Generische Implementierung)

Die Generische Implementierung dient der Umsetzbarkeit generischer Regeln. D.h. Regeln, die für mehr als ein Businessobjekt genutzt werden können.

Zu diesem Feature gibt es ein Tutorial

Der normale Regelkontext liefert immer das konkrete BO (z.B.

`Auftrag a = context.getBusinessObject(Auftrag.class)).`

Es gibt manchmal aber Anforderungen, die für eine Vielzahl von BOs gleichermaßen umgesetzt werden könnten, z.B. die Vergabe einer fortlaufenden Nummer beim Speichern. Die Umsetzung würde dann innerhalb einer Regel durch die Verwendung von instanceof gelöst werden. Etwa so:

Bisherige Implementierung ohne Generischen BOs

```
...
        final Modifiable<?> businessObject = context.getBusinessObject
(Modifiable.class);
        if (businessObject instanceof Geschaeftspartner) {
            Geschaeftspartner gp = ((Geschaeftspartner) businessObject);
        ...
    } else if (businessObject instanceof Ansprechpartner) {
            ...
    } else if (businessObject instanceof Angebot) {
            ...
    } else if (businessObject instanceof Auftrag) {
            ...
    }...
...
```

Mit Hilfe der Generischen Implementierung ist es nun möglich, die Gemeinsamkeiten mehrerer BOs in "generischen Objekten" zu abstrahieren und Regeln für solche GBOs zu implementieren.

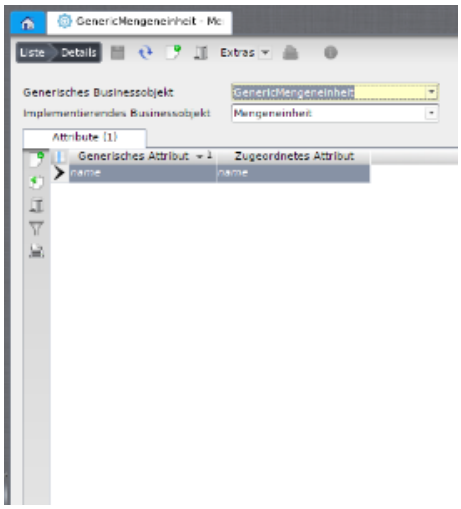
Konfiguration

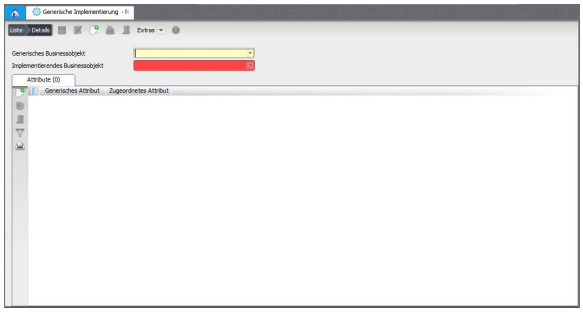
- Nummernkreise: Nummern müssen in der Regel für viele BOs generiert werden, ein entsprechendes Generisches Objekt könnte also z.B. „Nummerierbar“ heissen.
- Belege: In verschiedenen Belegarten, z.B. Angebot, Auftrag, Lieferschein, Rechnung, Gutschrift müssen unter Umständen in gleicher Art und Weise Netto- und Bruttosummen gebildet werden, entsprechende Generische Objekte könnten also z.B. „Beleg“ und „Belegposition“ heissen.

Voraussetzung ist die Konfiguration eines entsprechenden Generischen Businessobjekts im BO-Wizard und die Definition als solches über die Checkbox Generisches Businessobjekt.

Anschließend findet das Mapping der gewünschten BOs über diese Maske statt:

- **Generisches Businessobjekt:** Auswahl des BOs, das als generisches Businessobjekt definiert ist
- **Implementierendes Businessobjekt**: Auswahl des BOs, das in der generischen Implementierung verwendet werden soll
- **Attribute**: in diesem Unterformular wird die Zuordnung (das Mapping) der Attribute zwischen dem generischen BO und dem originalen BO vorgenommen. Ist ein Attribute im GBO als Pflichtfeld definiert, so muss ein Mapping angegeben werden. Es bedeutet nicht, dass das Attribute im originalen BO ein Pflichtfeld sein muss.





Begrifflichkeiten

***GenericBusiness*object** (in Analogie zum *Business*object) = Beschreibung eines BOs, das nicht für sich existiert, sondern dem Zweck dient, Gemeinsamkeiten mehrerer BOs zu abstrahieren, um die Anwendung einer Regel für all diese BOs zu erlauben (ohne in der Regel mit instanceof-Prüfungen arbeiten zu müssen).

Wrapperklasse = Automatisch generierte (und unabänderliche) Klasse, die die Attribute (bzw. deren Getter und Setter) eines GenericObject auf ein BusinessObject mappt.

Implementierendes BO = Hilfsbegriff für ein BO welches auf ein GBO gemappt ist. Dem Implementing BO sieht man nicht an, dass es ein Implementing BO ist.

Generische Regel = Hilfsbegriff für eine Regel, die mit GBOs arbeitet.

Aspekte

Generic Objects

- Generische Businessobjekte (GBO) können auf beliebig viele Businessobjekte (Implementierende BOs) gemappt werden.
- Ein Businessobjekt (Implementierendes BO) kann auf beliebig viele GBOs gemappt sein.
- Die Implementierenden BOs und deren Mappings sind Teil desselben Nuclets, das Mapping muss nicht separat zugewiesen werden.
- GBOs werden über den Businessobjekt erstellt. GBOs können Referenzfelder auf andere GBOs haben (siehe Beispiel Beleg und Belegposition im Tutorial).
- Das Mapping wird mithilfe der oben beschriebenen Maske erstellt
- GBOs können vom RuleContext nur mittels einer neuen Methode getGenericBusinessObject(Class GBOClass) abgeholt werden. Analog werden in GBOs Getter für die abhängigen GBOs benötigt (siehe Beispiel Beleg und Belegposition).
- GBOs werden nur durch Teile der Nuclos API unterstützt. Es sind insbesondere (trivialerweise) keine QueryProvider- oder StatemodelProvider-Aufrufe etc. für GBOs möglich.
- GBOs erhalten je Implementing BO einen Konstruktur, um ein Implementierendes BO in das Generische BO umwandeln zu können. Damit wird eine Verwendung außerhalb des RuleContextes möglich, wenn z.B. in einem Job ein normales BO zu einem GBO werden soll, um es einer Generic-Rule zu übergeben (MyGenericUtils.calculateSums(<GenericBusinessObject> gbo).
- GBOs können gelesen (Getter) und geändert (Setter), später ggf. auch gelöscht werden. Die entsprechenden Aufrufe werden an das zugrundeliegende, gemappte BO „delegiert“.

Generische Regeln

- müssen manuell an ein Implementierendes BO gehängt werden. Die Regelverwendungen sind damit im selben Nuclet wie die Implementierenden BOs.
- Ist das im Regelkontext befindliche BO kein Implementierendes BO des GBO, das versucht wird, aus dem Regelkontext zu holen, erfolgt eine Fehlermeldung zur Laufzeit.