

MODELLGETRIEBENE ENTWICKLUNG EINER MOBILEN APPLIKATION MIT JUSE4ANDROID

Jano Espenhahn, Tobias Franz and Franziska Krebs
Fachhochschule Brandenburg, Fachbereich Informatik und Medien
{espenhah, franzt, krebsf}@fh-brandenburg.de

Keywords: MDA, UML, USE, OCL, Android

Abriss: ein deutsches Abstract

Abstract: ein englisches Abstract

1 EINLEITUNG

Zitat Test (da Silva, 2014)

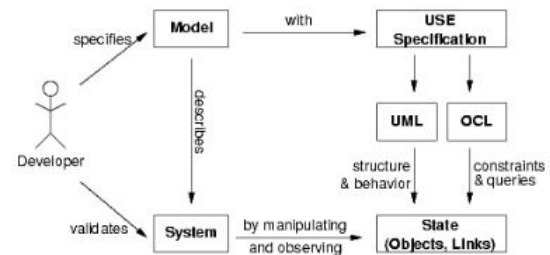
2 VORSTELLUNG USE

UML based Specification Environment (USE) wird zur Spezifikation von Informationssystemen verwendet und wurde an der Universität Bremen entwickelt. Neben dem Einsatz für Fallstudien, wird USE vor allem in der Lehre an Hochschule wie z. B. MIT, Cambridge, University of Edinburgh und University of Lisbon eingesetzt. USE basiert auf einer Teilmenge der Unified Modeling Language (UML) und der Object Constraint Language (OCL). Eine USE-Spezifikation besteht aus einer textuellen Beschreibung eines Modells, bei der Eigenschaften aus UML-Diagramm verwendet werden. Weitere Integritätsausdrücke für ein Modell können durch die OCL definiert werden. (?) Die OCL wird im späteren Kapitel (TODO) vorgestellt. Die nachfolgende Abbildung veranschaulicht den Workflow für eine USE-Spezifikation.

Ein Entwickler spezifiziert ein USE-Modell, welches ein System beschreibt und nutzt dafür UML- und OCL-Ausdrücke. Mithilfe von USE ist es ihm möglich die bestimmten Anforderungen an sein System auf Erfüllung mit dem Modell zu validieren.

2.1 Spezifikation

Die textuelle Beschreibung eines Modells mit USE beginnt immer mit der Definition eines Modell-



Workflow einer USE-Spezifikation (?)

Namens. In diesem Fall ist das *IceCream*. Im Anschluss folgen Klassendefinitionen mit ihren jeweiligen Attributen und Methoden. Im Beispiel hat die Klasse *Station* das Attribut *name* und die Operation *entries* ohne Übergabeparameter. Die nachfolgenden Code-Ausschnitte verwenden lediglich UML.

```
model IceCream

class Station
  attributes
    name : String
  operations
    entries() : Set(Entry) = self.records->asSet
end
```

Klassen können untereinander in Abhängigkeit stehen. Für diese Abhängigkeiten sind Assoziationen vorgesehen. Um eine Assoziation auszudrücken, wird zuerst eine weitere Klasse *Address* eingeführt.

```
class Address
  attributes
    street : String
    postCode : Integer
end
```

Für das dem Artikel zugrunde liegende Beispiel kann eine Station entweder eine oder keine Adresse haben.

```

association Station_Address between
  Station[ 1 ]
  Address[ 0..1 ] role place
end

```

Station_Address ist dabei der Name der Assoziation und das Attribut *place* nimmt in der Klasse *Station* die Rolle für die Adresse ein. Zum gesamten USE-Modell gehören weiterhin noch die Klasse *Entry* und die Assoziation *Station_Entry*.

```

class Entry
  attributes
    date      : CalendarDate
    target    : Integer
    actual    : Integer
    variance  : Integer
  operations
    variance() : Integer = actual - target
end

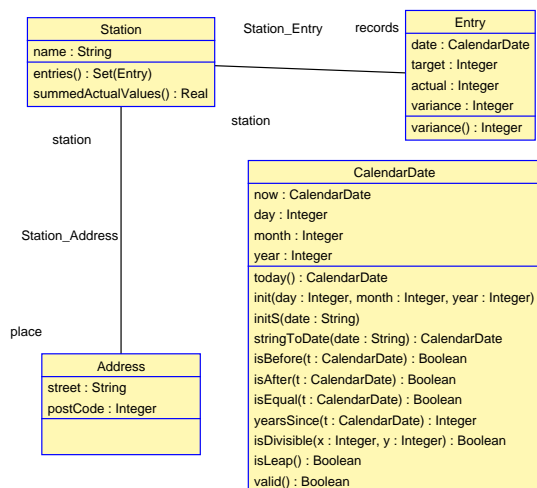
association Station_Entry between
  Station[ 1 ]
  Entry[ * ] role records
end

```

Zur Vervollständigung des Modells gehört außerdem eine aus der Arbeit (da Silva, 2014) entnommene Klasse *CalendarDate*.

2.2 Tool

Um eine Spezifikation auf nicht-formale Anforderungen zu validieren, kann ein Modell mithilfe des USE-Tools animiert werden. Direkt nach dem Import eines Modells erhält man vom Tool ein Feedback über die Validität der UML- und OCL-Definitionen. Neben der Validierung bietet das Tool weitere Möglichkeiten, wie z. B. die Visualisierung eines Klassen-, Sequenz- oder Objektdiagramms.



Klassendiagramm für das Beispiel

3 VORSTELLUNG OCL

4 JUSE4ANDROID

REFERENCES

da Silva, L. (2014). Model-driven generative programming for bis mobile applications. Master's thesis, ISCTE IUL University of Lisbon.

ANHANG

If any, the appendix should appear directly after the references without numbering, and not on a new page. To do so please use the following command: `\section*{APPENDIX}`