

# Protokoll Techreport-Treffen 24. Mai

Donnerstag, 24. Mai 2012  
10:12

## 1. Pieter Van Gorps Beitrag

Pieter soll ein, zwei Absätze zu seinem Kram schreiben. Wir bauen das dann in den Abschnitt über SD-Erweiterungen ein. Markus sagt ihm Bescheid.

## 2. Meta-Modell-Änderungen

- a. Konform zur UML benennen wir Start Nodes ab sofort in Initial Nodes und Stop Nodes in Final Nodes um. Final Nodes werden auch auf Meta-Modell-Ebene in zwei Elemente geteilt: Flow Final Node und Activity Final Node.
- b. Korrektur zum letzten Protokoll: ObjectSetVariable wird in CollectionVariable (statt ObjectCollection) umbenannt. Das ist konsistenter zu ObjectVariable.
- c. In dem Zusammenhang wird die SetSizeExpression erst mal auf Eis gelegt. Man kann das gleiche auch mit OCL ausdrücken. Die Klasse bleibt aber im Metamodell.
- ? d. Binding Semantics und Binding Operator sind keine so tollen Namen für die dahintersteckenden Konzepte. Es gibt noch keine brillanten Vorschläge für eine Umbenennung, aber bis zum nächsten Treffen sollten wir uns Gedanken dazu machen.

Markus kümmert sich um die Umbenennungen im Dokument.  
Ingo übernimmt die Änderungen am Meta-Modell.

## 3. Konkrete Syntax

- a. **Pattern Constraints:** Das Compartment für die Pattern Constraints soll an den unteren Rand der Activity Nodes verlegt werden. Das reflektiert besser, dass die Constraints erst nach dem Matching der Objektstruktur ausgewertet werden.
- b. **Link Order Constraints:** Direct\_successor links erhalten das Label {next} statt {direct\_successor}. Die abstrakte Syntax bleibt so wie sie ist.

## 4. Semantikfragen

- a. **Pattern Constraints:** Sollen die Pattern Constraints in der Section in der sie erklärt werden frei rumfliegen (denn Activity Nodes in denen sie eingeordnet sind, gibt es da noch nicht)?  
Antwort: Ja.
- b. **Collections:** Wir unterstützen der Einfachheit halber vorerst nur (Ordered) Sets und Lists, ungeordnete Collections gibt's nicht.
  - i. Binding Semantics und Operators für Collections:  
Collections können nicht negativ oder optional sein.  
Destroy geht, Create ist verboten.
  - ? ii. Sollen wir ein Attribut einführen, das angibt ob die Collection mindestens ein Element enthalten MUSS? (Das müsste man sonst immer per Object Constraint ausdrücken.)
- c. **Link Position Constraints**
  - i. Index gibt's nicht wegen semantischer Unklarheiten beim Einfügen von mehreren Elementen. Das erklären wir kurz im Dokument.
  - ii. Link Position Constraints an Optional Create Links (s. Figure 3.15): Wir fügen das Element gemäß dem Position Constraint ein. Bei Unique-Referenzen, bedeutet das eine Verschiebung des Elements. Bei Nicht-Unique-Referenzen existiert das Element dann zweimal in der Liste.
  - iii. Link Position Constraints an Negative Links (s. Figure 3.16): Ein Link Position Constraint an einem negativen Link bedeutet, dass das Objekt nicht an der durch das Constraint bezeichneten Position stehen darf (von Christian vorgeschlagene Semantik 1). Das bedeutet eine leichte Abwandlung der normalen Semantik von Negative Links.
- d. **Link Order Constraints**
  - i. Link Order Constraints und Binding Operators 1: Links, die per direct oder indirect\_successor link constraint verbunden sind, dürfen sowohl erzeugt, als auch gelöscht werden. Die Semantik für indirect successor links, die erzeugt werden ist, dass das neue Element direkt hinter dem anderen eingefügt wird (also kein Unterschied zu direct successor).
  - ii. Link Order Constraints und Binding Operators 2: Wenn ein Element indirekter Nachfolger von zwei andern Elementen ist, und erzeugt werden soll, dann wird das hintere der beiden Elemente bestimmt und das neue direkt dahinter eingefügt.
  - iii. Link Order Constraints und optionale Links: Optional Links sind sowohl für direct, als auch für indirect successors erlaubt. Ein optional create bei einem indirect successor bedeutet: Falls Element 2 in der Liste nicht irgendwo hinter Element 1 vorhanden ist, dann wird es (direkt) dahinter eingefügt.
  - iv. Link Order Constraints und negative Links: Negative Links sind sowohl für direct, als auch für indirect successors erlaubt. Die Semantik entspricht der Semantik 1, die von Christian vorgeschlagen wurde: Ein negativer Link mit einem Link Order Constraint bedeutet, dass das zweite Element kein (in-)direkter Nachfolger des ersten Elements ist.

e. **Ergänzung in Abschnitt 3.2.4.3 Binding Operators**

Es kann passieren, dass das Matching eines Patterns klappt, aber die Transformation schiefgeht. Das ist z.B. der Fall, wenn widersprüchliche Link Order Constraints an create-Links stehen. Dann würde erst beim Einfügen des neuen Elements festgestellt, dass nicht alle Constraints erfüllt werden können. In dem Fall gibt's eine Exception vom Interpreter. Das Story Diagramm wird dann nicht irgendwie vernünftig weiter ausgeführt.

f. **Activity Final Nodes**

Beim Erreichen eines Failure Final Nodes gibt's kein Rollback. Der Entwickler muss damit rechnen, dass eine fehlgeschlagene Anwendung trotzdem Seiteneffekte haben kann.

5. **Review**

Folgende Teile sind aktuell für v0.2 fertig und können bei Gelegenheit gereviewt werden:

3.2.9: Maybe-Links

4. Complete Example

6. **Nächstes Treffen:**

Donnerstag, 14.6., 10 Uhr. (Dietrich ist im Urlaub)