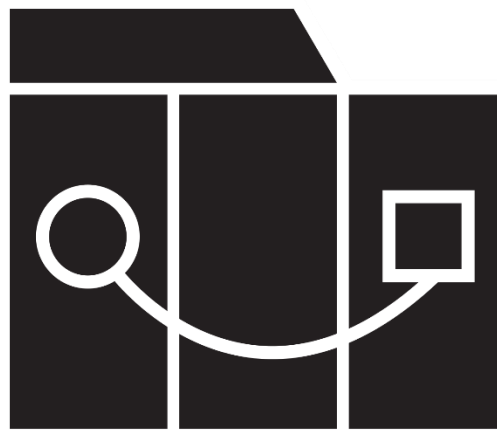


Anforderungen an den Story Pattern Editor von SE2 WS1415



SOFTWARE ENGINEERING
UNIVERSITÄT KASSEL

Version	Datum	Änderung
1.0	12.01.15	init
2.0	24.02.15	1.2.1, 2.1 Neu: 1.13, 1.14, 4

1 Story Pattern Editor

Es soll mithilfe des Graphiti Frameworks ein graphischer Editor für Story Pattern erstellt werden.

1.1 UI: Attribute Condition

Unterstützung für folgende Attribute Conditions

1.1.1 Equals Operator

name == „Alice“

1.1.2 Not Equals Operator

name != „Bob“

1.1.3 Less Operator

age < 23

1.1.4 Less Or Equals Operator

age <= 23

1.1.5 Greater Operator

age > 23

1.1.6 Greater Or Equals Operator

age >= 23

1.1.7 RegEx Operator

name == [a-zA-Z]+

1.1.8 Interval Operation

age > 23 < 42

1.2 UI: Direct Edit Feature

Siehe:

http://help.eclipse.org/luna/index.jsp?topic=%2Forg.eclipse.graphiti.doc%2Fresources%2Fdocu%2Ffw%2Fdirect-editing-feature.htm&cp=29_1_3_9

1.2.1 Completion

Wenn in dem Story Pattern Diagram, bei einem Pattern bspw. alice : Person angegeben wurde, soll bei der nächsten Hinzufügen eines Pattern bei der Eingabe des Typs, passende Typen vorgeschlagen werden. Bei z.B. bob : P soll Person vorgeschlagen werden. Die Vorschläge sollen aus dem Buildpath des Eclipse Projektes geladen werden.

Siehe: IDirectEditing.completeValue(..)

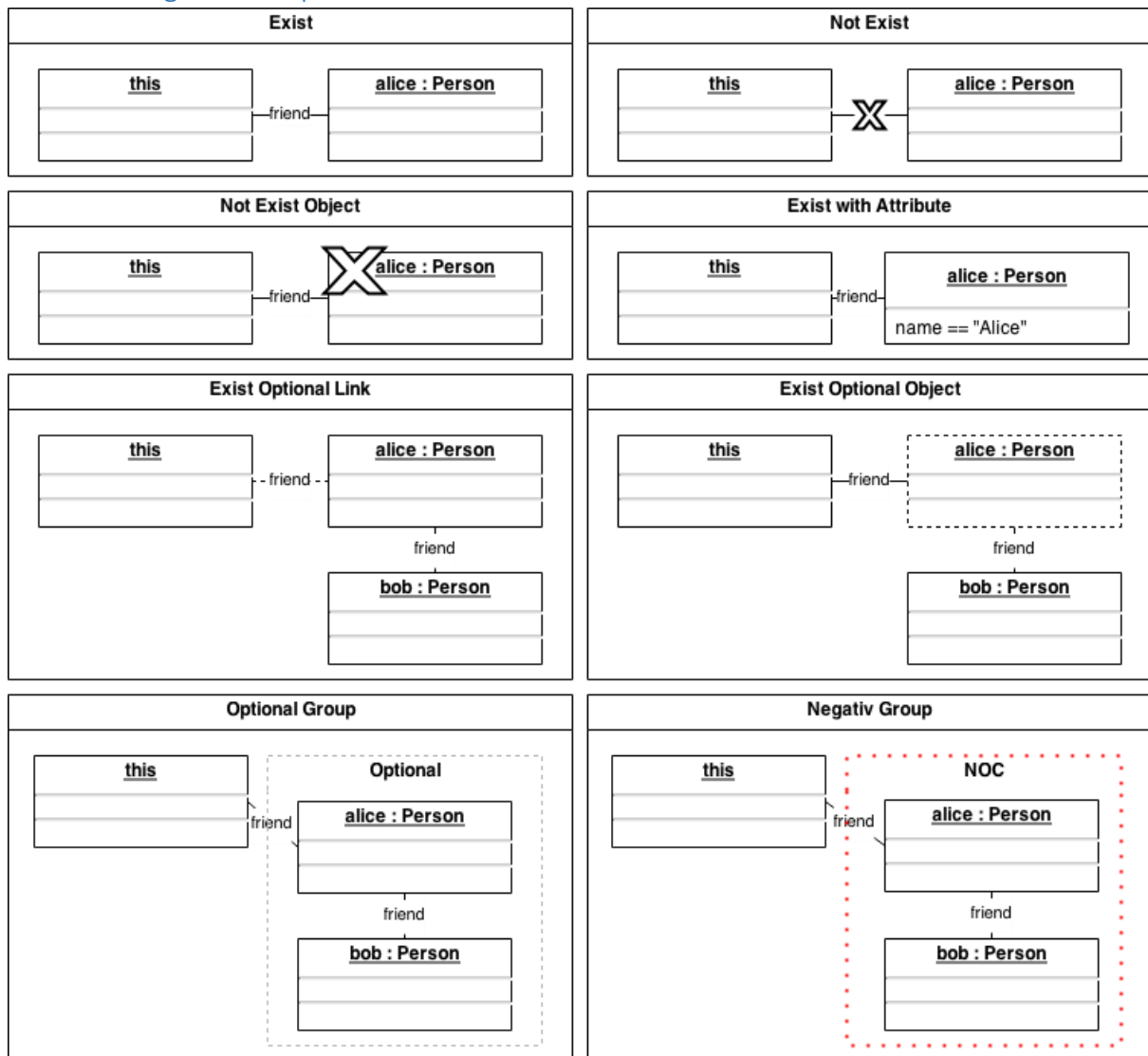
1.2.2 add Attribute/Operation via Objectname

Wenn der Name eines Patterns geändert wird, soll geprüft werden, ob es sich beim neuen Namen um eine Attribute Conditions handelt. Sollte dies so sein, soll der Name nicht geändert werden sondern die Condition dem Pattern hinzugefügt werden.

1.3 UI: New Story Diagram Wizard

Siehe: <http://www.vogella.com/tutorials/EclipseWizards/article.html>

1.4 UI: Negativ Group Condition



1.5 UI: Optional Group Condition

(Bild siehe 1.4)

1.6 UI: Attribute Condition

(Bild siehe 1.4)

1.7 UI: Exist Optional Object Condition

(Bild siehe 1.4)

1.8 UI: Exist Optional Link Condition

(Bild siehe 1.4)

1.9 UI: Not Exist Object Condition

(Bild siehe 1.4)

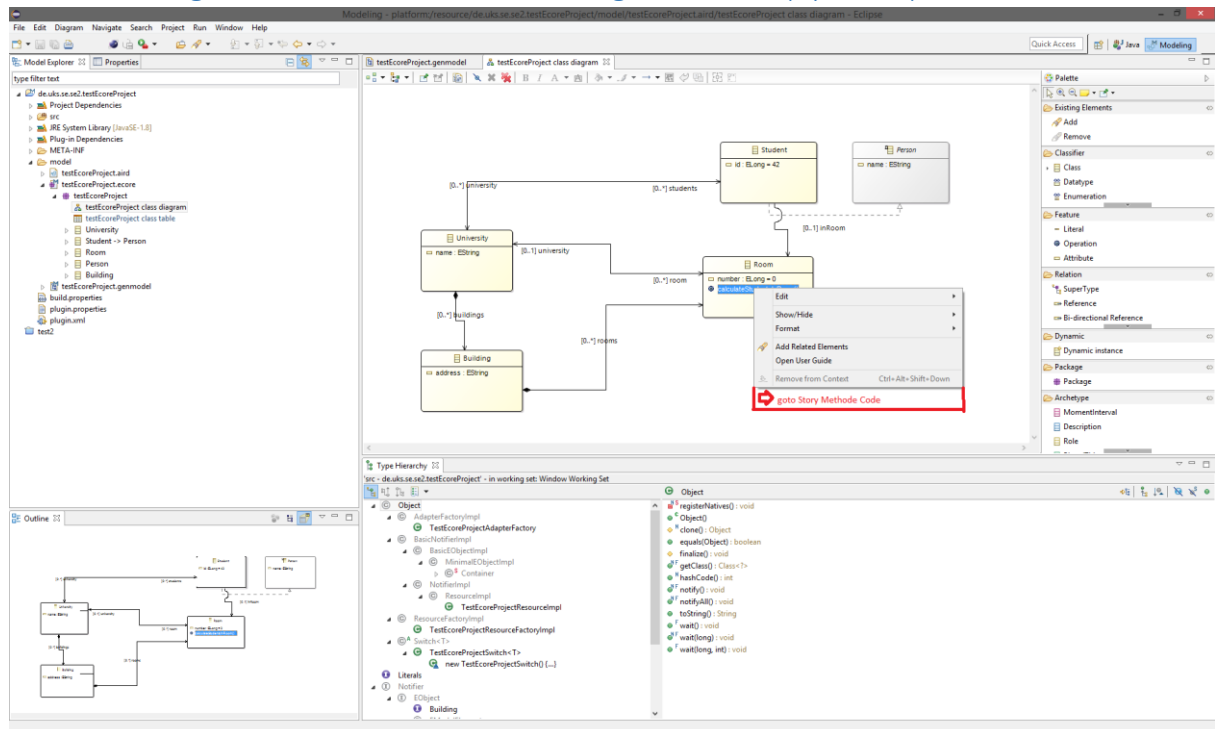
1.10 UI: Not Exist Link Condition

(Bild siehe 1.4)

1.11 UI: Exist Link Condition

(Bild siehe 1.4)

1.12 UI: Integrate in EcoreTools 2 Class Diagram Editor (optional)



Optionale Aufgabe

1.13 UI: Attribute assignment

Einem Attribut soll ein neuer Wert zugewiesen werden können: name := „Alice“. Dieser soll im generierten Code nach dem Matching gesetzt werden.

1.14 UI: Create, Delete Operation

Es soll möglich sein, einem PatternObject, eine Connection und einer Group eine Create bzw. einer Delete Operation zuzuweisen. Diese soll im generierten Code nach dem Matching berücksichtigt werden.

2 Story Pattern Model

Es soll ein geeignetes EMF Metamodell erstellt werden.

2.1 MM: integrate class reference in story pattern model

Im Metamodell muss eine Referenz zur EClass oder EJavaClass eingeführt werden können. Die möglichen EClasses bzw. EJavaClass sollen aus dem Buildpath des Eclipse Projektes erfasst werden.

3 Model to Code

Codegenerator zur Übersetzung der gezeichneten Story Pattern nach SDMLib Code.

Erst soll die Anwendungstelle gesucht werden (Matching) dann sollen Änderungen durchgeführt werden (Attribute assignments & Create, Delete Operations)

4 Test

Es soll ein SWTBot test erstellt werden, der Mindestens den eigenen Wizard benutzt um ein neues Story Pattern Diagramm erzeugt. Auf dieses Diagramm zwei PatternObjects legt, diesen beiden Objekten ein Attribute zuweist und diese anschließend einer Connection verbindet. Das Diagramm soll vom Test anschließend gespeichert werden und der Codegenerator angestoßen werden.