

Projektentwicklung mit Together R2 for Eclipse

Im Rahmen des Labors für Softwaretechnik soll eine objektorientierte Softwareentwicklung durchgeführt werden. Als Entwicklungswerkzeug wird 'Together R2 for Eclipse' von Borland eingesetzt, welches die grafische Spezifikationssprache UML umsetzt.

Um die einzelnen Entwicklungsphasen zu verdeutlichen und um die Handhabung des Werkzeugs zu erläutern, wird ein Beispielprojekt, die Entwicklung eines Kniffelspiels, durchgeführt.

Im vorliegenden Dokument wird die Handhabung von Together in groben Zügen an Hand von Beispielen dargestellt, teilweise sind auch Verweise auf die Dokumentation eingefügt.

2	Design.....	2
2.1	Bedienkonzept erarbeiten	2
2.2	System Use Case Modell erstellen	2
2.2.1	System Use Case Diagramm	3
2.2.2	Textuelle Beschreibung der System Use Cases	3
2.2.3	Beschreibung der System Use Cases durch Aktivitätsdiagramme	3
2.2.4	Systemoperationen.....	3
2.3	Zustandsdiagramm erstellen	3
2.4	Die Applikationsschicht	5
2.4.1	Detailliertes Sequenzdiagramm erstellen	5

2 Design

2.1 Bedienkonzept erarbeiten

Bedienkonzept erarbeiten heißt sich zu überlegen wie muss die Oberfläche, d.h. die verschiedenen Fenster, zu meinem Spiel aussehen.

Beispiel:

The screenshot shows a Kniffel game window with a yellow background. At the top, there's a title bar 'Kniffel' and a close button. Below it are two tabs: 'Spiel' and 'Hilfe'. The main area is divided into several sections. On the left, there's a list of dice counts from 1 to 6, each with a corresponding number of dots and a 'zählen' button. Below this is a section for 'Oben gesamt' and 'Bonus (+35 Pkt., ab 63Pkt.)'. Further down is a list of game rules with their respective point values: Dreierpasch (25 Punkte), Viererpasch (30 Punkte), Full House (40 Punkte), Kleine Straße (50 Punkte), Große Straße (100 Punkte), Kniffel (100 Punkte), Chance, and ExtraKniffel *. On the right side, there's a section for 'Runde: 7', 'Spieler: Name5', and 'Wurf: 2'. Below this are three dice icons, each with a 'Halten' button. At the bottom right is a large 'Würfeln' button. The bottom of the window has a section for 'unten gesamt' and 'Gesamte Punktzahl:'. The player names are listed at the top: Name1, Name2, Name3, Name4, Name5, Name6.

Das vollständige Bedienkonzept ist im Design-Dokument "Kniffel-Design.doc" zu finden.

2.2 System Use Case Modell erstellen

Für das Design das Package Design anlegen. Für eine ausführliche Beschreibung siehe das Kniffel Design Dokument.

Package Design in Projekt anlegen

Select Projektknoten, rechtsklick → new → Package
Name eingeben, z.B. 'Design'

2.2.1 System Use Case Diagramm

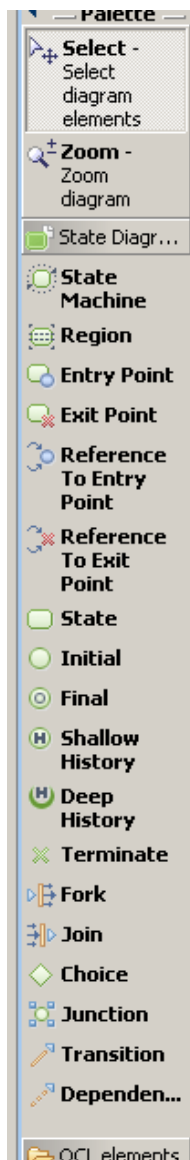
2.2.2 Textuelle Beschreibung der System Use Cases

2.2.3 Beschreibung der System Use Cases durch Aktivitätsdiagramme

2.2.4 Systemoperationen

2.3 Zustandsdiagramm erstellen

Select Packageknoten 'Design', rechtsclick → New Diagram → State Machine

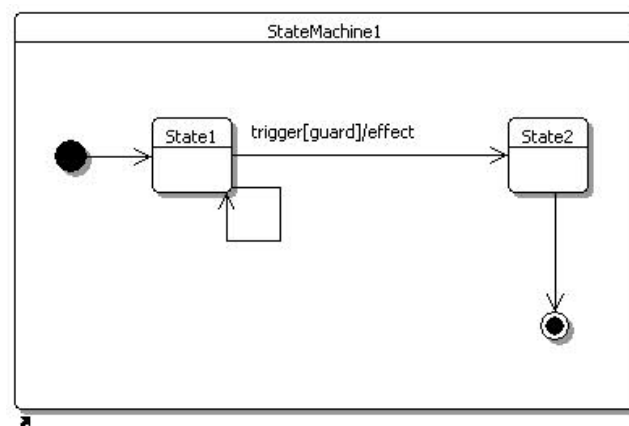


Zeichnen

- der **State Machine**
- des **Initial** – Zustands
- der **States**
- des **Final** – Zustands

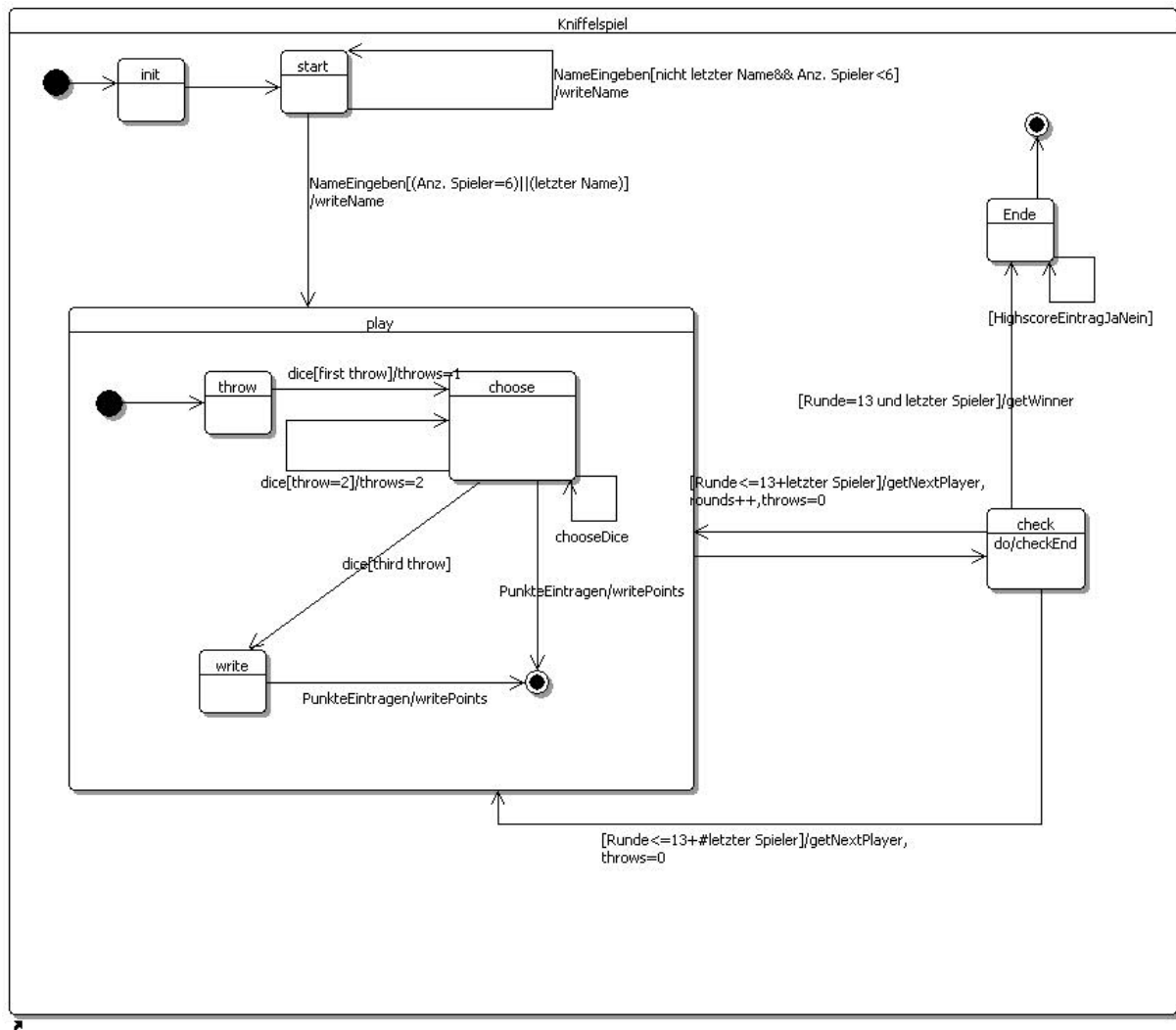
Verbinden der **States** durch **Transitions**

Beschriften der Transitionen



Die Trigger entsprechen den Systemoperationen.
Guard ist die Bedingung unter der der Zustandsübergang stattfindet.
In effect können nur schon vorhandene activites eingetragen werden.

Beispiel



2.4 Die Applikationsschicht

Detailliertes Sequenzdiagramm erstellen

Ausgehend von den Systemoperationen wird für jede Systemoperation ein detailliertes Sequenzdiagramm angelegt.

Auswahl der Systemoperation im Model Navigator oder im Klassendiagramm

Select Operation rechts-click auf diese Operation → Generate Sequence Diagram.

In diesem Diagramm kann nun die Operation mit Inhalt gefüllt werden.

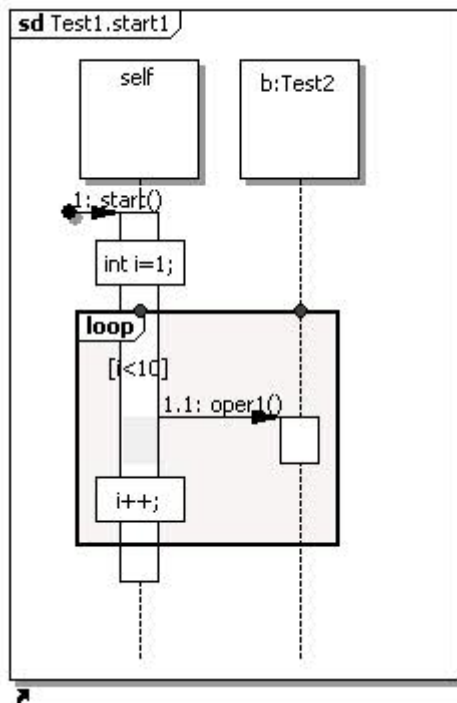
Erzeugen von Code

Select Interaktion, rechts-click auf die Interaktion → Generate Implementation

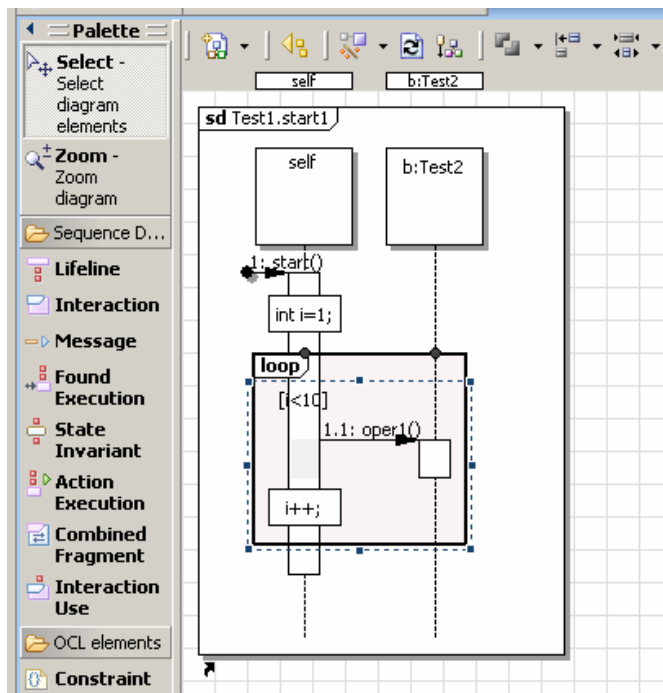
Hierdurch wird der im Diagramm dargestellte Ablauf in Code umgesetzt.

Umsetzung der verschiedenen Combined Fragments

loop



```
public void start() {
    // generated by Together
    Test2 b = null;
    int i=1;
    while (i<10) {
        // Message #1.1 to b:Test2
        b.oper1();
        i++;
    }
}
```



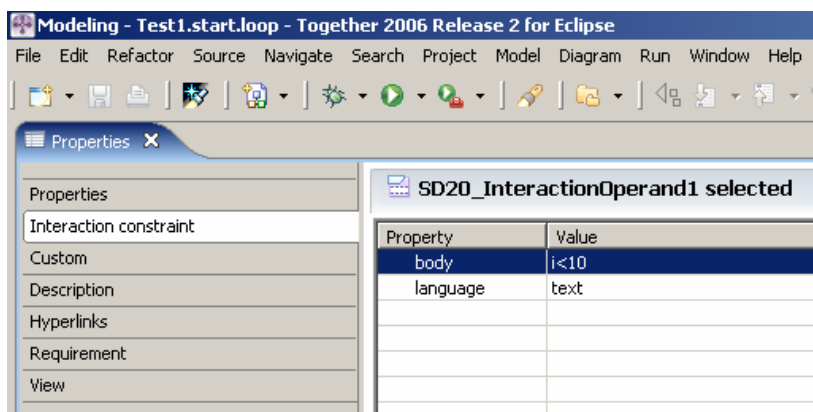
Der Eintrag der Bedingung [i<10]

Selektieren des Interaction Operands

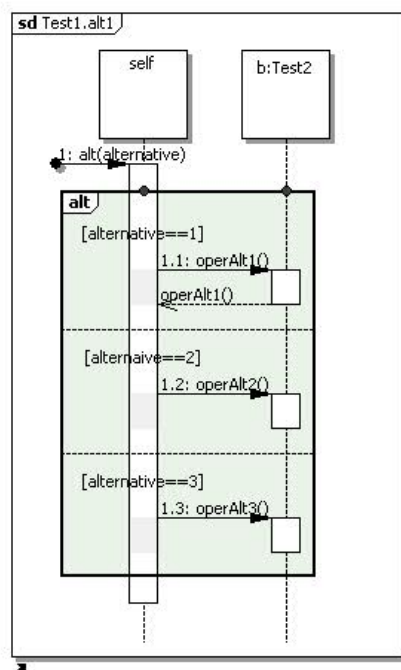
In den Properties erscheint der Tab „Interaction constraint“

language text
body i<10
eintragen

Das Element “int i= 1;” ist ein Action Execution Element



alt

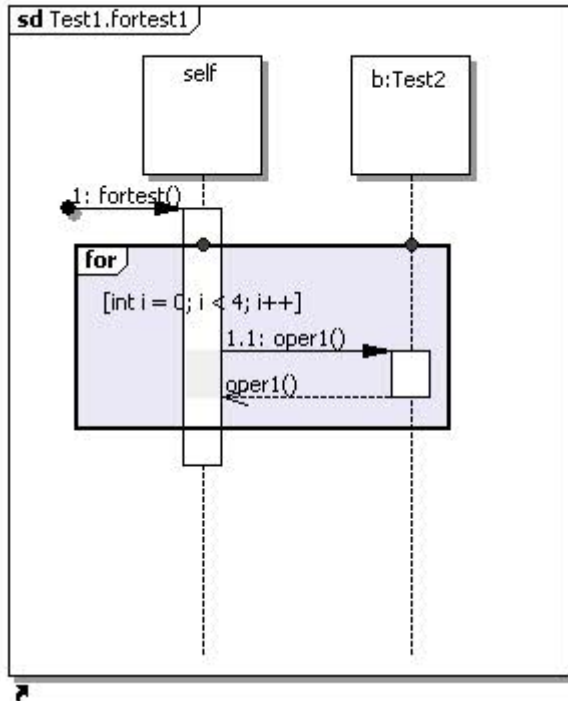


```

public void alt(int alternative) {
    // generated by Together
    Test2 b = null;
    if (alternative==1) {
        // Message #1.1 to b:Test2
        b.operAlt1();
    }
    else if (alternative==2) {
        // Message #1.2 to b:Test2
        b.operAlt2();
    }
    else if (alternative==3) {
        // Message #1.3 to b:Test2
        b.operAlt3();
    }
}
  
```

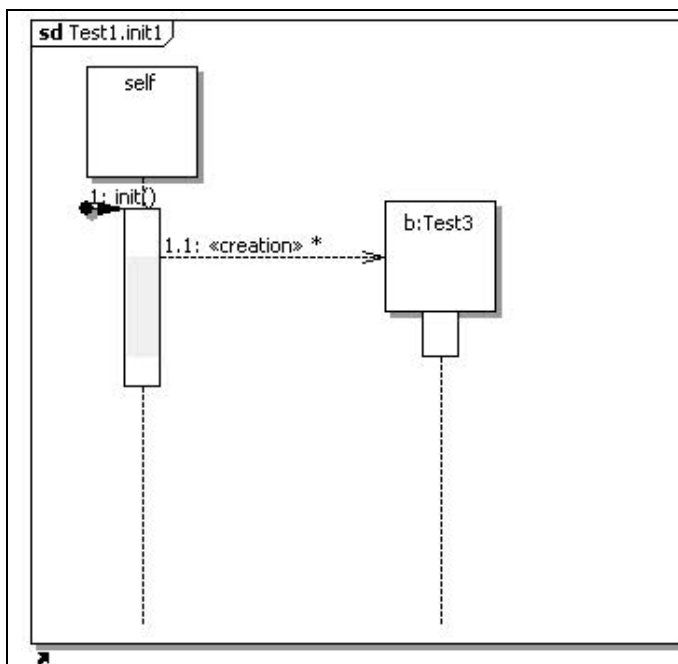
For

Um die Reply Message anzuzeigen ist in den Properties der Operation bei „show reply message“ ein Haken zusetzen .



```
public void fortest() {
    // generated by Together
    Test2 b = null;
    for (int i = 0; i < 4; i++) {
        // Message #1.1 to b:Test2
        b.oper1();
    }
}
```

Creation



```
public void init() {
    // generated by Together
    Test3 b = null;
    // Message #1.1 to b:Test3
    b = new Test3();
}
```

