

Softwaretechnik II, Sommersemester 2010

Übungsblatt 6: Transformationswerkzeuge für Modelle

Besprechung: 15.07.2010

8. Juli 2010

Aufgabe 6.1

Gegenstand dieser Übungsaufgabe ist eine einfache Variante des Spiels “PacMan”. Die Spielfiguren des Spiels, *PacMan* und *Geist*, bewegen sich auf fest definierten *Feldern*. Für alle Felder sind deren benachbarte Felder statisch definiert und können sich während des Spielablaufs nicht mehr ändern. Die Spielfiguren PacMan und Geist können sich über die Nachbarschaftsbeziehungen von Feld zu Feld bewegen. Auf Feldern können sich zudem *Murmeln* befinden, die von PacMan eingesammelt werden können. PacMan selbst wiederum kann durch Geister gefressen werden. Eine Repräsentation der beschriebenen Konzepte ist dem in Abb. 1 dargestellten Typgraphen zu entnehmen. In der Terminologie der modellbasierten Entwicklung wird dieser auch als Metamodell bezeichnet.

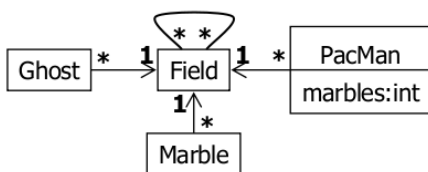


Abbildung 1: Statische PacMan-Konzepte: Typgraph (Metamodell)

Nutzen Sie das *Eclipse Modeling Framework* (EMF) sowie das Graphtransformationssystem *EMF-Henshin*, um den Ablauf des PacMan-Spiels mittels der Anwendung von Graphtransformationen zu simulieren. Das Transformationssystem soll dabei folgende Spielregeln umfassen:

- PacMan bewegen (**movePM**): Hierbei wird PacMan von seinem aktuellen Feld aus auf ein benachbartes Feld verschoben.
- Murmel einsammeln (**collect**): Auch bei der Anwendung dieser Regel wird PacMan von seinem aktuellen Feld aus auf ein benachbartes Feld verschoben. Auf diesem befindet sich jedoch zusätzlich eine Murmel, die im Zuge der Regelanwendung von dem Feld entfernt wird. Gleichzeitig wird die Anzahl der von PacMan eingesammelten Murmeln um eins inkrementiert.
- Geist bewegen (**moveGhost**): Ebenso wie PacMan kann auch ein Geist von seinem aktuellen Feld aus auf ein benachbartes Feld verschoben werden.
- PacMan fressen (**kill**): Wird ein Geist auf ein benachbartes Feld verschoben, auf dem sich PacMan befindet, so wird dieser gefressen (gelöscht).

Lösungshinweise

1. Hinweise zum Download der benötigten Software (Eclipse + entsprechende Plugins) finden Sie unter <http://pi.informatik.uni-siegen.de/Mitarbeiter/kehrer/lehre/ss10/st2/>.
2. Beginnen Sie zunächst mit der Umsetzung des in Abb. 1 dargestellten Typgraphen in EMF-Ecore. Beachten Sie dabei insbesondere die technische Rahmenbedingung, dass für jede Klasse eines Ecore-Modells eine Container-Klasse definiert ist.
3. Formulieren Sie die Transformationsregeln mittels des in der Vorlesung vorgestellten, grafischen Editors (Henshin Diagramm). Vergessen Sie nicht, den zuvor definierten Typgraphen zu importieren.
4. Erzeugen Sie sich eine oder mehrere dynamische Instanzen Ihres Ecore-Modells, wie Sie es in den EMF-Übungen bereits mehrfach durchgeführt haben. Diese können Sie anschließend in Schritt 6 als Testdaten (Ausgangsgraphen) für die Anwendung der Transformationsregeln nutzen.
5. Programmieren Sie einen einfachen Testtreiber zur Simulation des Spiels. Dieser lädt das Transformationssystem und führt die anwendbaren Regeln in einer zufälligen Abfolge auf den zuvor in Schritt TODO definierten Testgraphen aus.
6. Als Anschauungsbeispiel für die technische Umsetzung in EMF-Henshin dienen die in der Vorlesung vorgestellten *Henshin-Examples*. Importieren Sie sich dazu das Plugin “org.eclipse.emf.henshin.examples” in den Arbeitsworkspace Ihrer gemäß Schritt 1 konfigurierten Eclipse-Umgebung (File > Import > Plug-Ins and Fragments > Projects with source folders > org.eclipse.emf.henshin.examples).

In der darin enthaltenen Java-Klasse `sierpinski.SierpinskiBenchmark` finden Sie konkrete Hinweise für die Implementierung des Testtreibers.

Viel Erfolg!