



Statecharts versus Monaco

Ein Vergleich

Diplomarbeit am

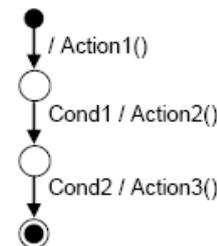
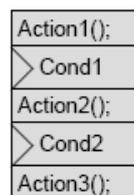
Institut für Systemsoftware

Aufgabenstellung

Am Christian Doppler Labor für Automated Software Engineering wurde zusammen mit der Firma KEBA eine Domänenspezifische Sprache für die Automatisierungstechnik entwickelt (Monaco - **M**odeling **N**otation for **A**utomation **C**ontrol). Monaco erlaubt es, die Hardware-Komponenten einer Maschine durch Software-Komponenten zu modellieren, die jeweils auf Kommandos reagieren sowie Daten liefern und Ereignisse auslösen können. Damit lassen sich Maschinensteuerungen auf natürliche Weise in Software abbilden.

Ein Monaco-Programm ist eine textuelle Darstellung eines Statecharts [HKKR05, Har87]. Es besteht aus Arbeitsschritten (z.B. Berechnungen und Zuweisungen) und WAIT-Anweisungen, in denen es auf ein Ereignis oder eine boolesche Bedingung wartet. Im Zuge des Monaco-Projekts wurde auch eine auf Eclipse basierende visuelle Entwicklungsumgebung (IDE) geschaffen, in der man Monaco-Programme in textueller und grafischer Form editieren kann, wobei die beiden Darstellungen jederzeit umgeschaltet werden können. Als grafische Form wurde eine Notation verwendet, die Ähnlichkeiten zu Struktogrammen hat. Monaco-Programme, Monaco-Struktogramme und Statecharts sind ineinander transformierbar, wie die folgende Abbildung zeigt:

```
BEGIN  
  Action1();  
  WAIT Cond1;  
  Action2();  
  WAIT Cond2;  
  Action3();  
END
```



Monaco-Programm

Monaco-Struktogramm

State Chart

In dieser Diplomarbeit soll, aufbauend auf der vorangegangenen Projektarbeit, eine Umsetzung von Monaco-Programmen in Statecharts realisiert und daran anschließend ein Vergleich dieser beiden Notationen angestellt werden. Der Vergleich beinhaltet eine Evaluierung der beiden Notationen hinsichtlich Verständlichkeit, Übersichtlichkeit und Benutzerfreundlichkeit, sowie ein grundsätzlicher Vergleich der Programmierung in Monaco und der Modellierung in Standard Statechart-Tools.

Literatur

[HKKR05] Hitz, M., Kappel, G., Kapsammer, E., Retschitzegger, W.: UML@Work. 3. Auflage, dpunkt.verlag, 2005

[Har87] Harel, D.: Statecharts: A Visual Formalism for Complex Systems. Sci. Comput. Prog. 8, 231-274, 1987