



Grafische Programmierung von Maschinensteuerungen durch Endbenutzer

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. Monaco-Programme sollen allerdings auch von Endbenutzern an der Maschine erstellt und geändert werden. Diese Endbenutzer sind keine Programmierer und benötigen daher eine Notation, die ihrer gewohnten Denkweise entgegenkommt.

Ziel dieses Projekts ist es daher, eine grafische Darstellung der zu programmierenden Maschine so mit Monaco-Komponenten zu hinterlegen, dass man Programme durch Anklicken der entsprechenden Maschinenteile und Auswahl der gewünschten Kommandos (d.h. Routinen) zusammenstellen kann. Dazu sind folgende Aufgaben zu lösen:

1. Erstellen Sie einen Grafikeditor, mit dem man Linien, Rechtecke, Kreise, Texte und Polygone zeichnen, färben und füllen kann. Es damit soll möglich sein, eine Maschine schematisch darzustellen. Es sollen auch Bilder (GIF, JPG, u.dgl.) mit einbezogen werden können. Mehrere Grafikelemente sollen zu einer Figur gruppiert werden können.
2. *Modelliermodus*: Es soll möglich sein, einzelnen Figuren oder Figurengruppen der Maschine Monaco-Komponenten zuzuordnen. Greifen Sie auf die internen Datenstrukturen der Monaco-IDE zu und bieten Sie alle darin in einem Monaco-Programm enthaltenen Komponenten in einem Teilfenster des Editors an, aus dem man dann einzelne Komponenten auswählen und Figuren der Zeichnung zuordnen kann. Es soll auch nach der Zuordnung ersichtlich sein, welche Komponenten welchen Figuren zugeordnet wurden.

3. *Programmiermodus*: Wenn der Benutzer im Programmiermodus auf eine Figur klickt, der eine Monaco-Komponente zugeordnet ist, soll ein Menü aufklappen, das alle Routinen, Funktionen und Ereignisse dieser Komponente anzeigt. Der Benutzer soll eine dieser Routinen auswählen können, wodurch sie an der jeweiligen Einfügeposition in das Monaco-Programm übernommen wird. Das entstehende Programm soll in einem Nebenfenster des Editors textuell dargestellt werden. Die Einfügeposition soll vom Programmierer in diesem Textfenster als Cursor oder Selektion gesetzt werden können. Wenn eine Routine Parameter hat, so sollen zunächst im Text Platzhalter erzeugt werden, die dann entweder textuell editiert oder durch Selektion einer Funktion aus einer Komponente wie oben beschrieben ausgewählt werden können. Das Textfenster stellt genau genommen eine einzelne Routine dar, deren Quelltext mit Hilfe der grafischen Maschinendarstellung editiert wird. Daher soll es möglich sein, für eine bestimmte Monaco-Komponente eine neue Routine anzulegen bzw. eine bestehende auszuwählen. Der Quelltext dieser Routine soll dann im Textfenster angezeigt werden und grafisch editiert werden können. Dabei wäre es wünschenswert, wenn nur jene Figuren in der Zeichnung aktiv wären, die in der Monaco-Komponente benutzt werden, zu der die editierte Routine gehört.

Experimentieren Sie mit diesem Werkzeug und versuchen Sie, sich in die Denkweise eines Endbenutzers zu versetzen. Versuchen Sie, den Benutzer bestmöglich zu unterstützen und ihm gleichzeitig die Erstellung vollständiger Monaco-Programme zu ermöglichen.

Das Werkzeug soll in Eclipse unter Java erstellt werden. Studieren Sie die Sprache Monaco und ihren Compiler. Das Frontend des Monaco-Compilers und seine internen Datenstrukturen sollen für Ihren Editor wiederverwendet werden. Achten Sie bei der Implementierung des Werkzeugs auf guten Programmierstil und ausführliche Kommentierung, damit es später auch von anderen Personen gewartet werden kann.

Betreuer: o.Univ.-Prof. Dr. Hanspeter Mössenböck

Beginn: Oktober 2007