

Übungsblatt 5

Revision: 1.3

In der Vorlesung wurde der in Abbildung 1 dargestellte Workflow zur Erzeugung von ausführbarem Code aus DSL-Modellen vorgestellt. Das Ziel dieses Übungsblattes ist es, den in dieser Abbildung grau markierten Teil umzusetzen.

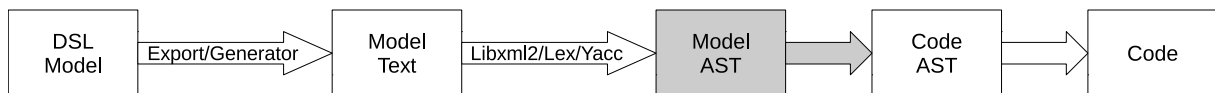


Abbildung 1: Workflow: vom DSL-Modell zum ausführbaren Code

Beim Model-AST handelt es sich um eine vom verwendeten Case-Tool unabhängige Zwischendarstellung von Modellen, die - einmal aus der textuellen Repräsentation eines Modells erzeugt - für weitere Transformationen genutzt werden kann, wobei in diesem Übungsblatt insbesondere Transformationen zum Code-AST von Interesse sind.

Da die Umsetzung des Model-AST nicht unabhängig vom betrachteten Metamodell sein kann, soll im Folgenden vom von Ihnen in Übungsblatt 2 entwickelten Metamodell für *Komponentenhierarchien mit hierarchischen Timed Automata* (im Folgenden: HTA-Modelle) ausgegangen werden.

Aufgabe 1: Implementieren Sie alle für die Repräsentation von HTA-Modellen als Model-AST erforderlichen Klassen sowie eine Transformation, die es erlaubt, ein Model-AST in den zugehörigen Code-AST zu übersetzen. Verwenden Sie das Composite-Pattern zur Abbildung von Komponenten- und Locationhierarchien der HTA-Modelle im Model-AST. Wenden Sie bei der Implementierung der Code-AST-Transformation das Visitor-Pattern an.

Die in der Vorlesung bereits erarbeiteten Teile dieser Aufgabe können von der SeS-Webseite heruntergeladen werden. Die (unvollständige) Implementierung soll geeignet ergänzt/angepasst werden, wobei auch diesmal die zusätzlichen Hinweise von Übungsblatt 4 zu berücksichtigen sind.

Aufgabe 2: Testen Sie Ihre Implementierung, indem Sie geeignete Model-AST-Instanzen erzeugen und diese in Code-AST transformieren.

Abgabe: 01.02.2011 bis 16:00 Uhr