

## Übungsblatt 6

Revision: 1.0

In der Vorlesung wurde der in Abbildung ?? 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. Die Implementierung des blau eingefärbten Teils wird bereitgestellt, so dass nach Bearbeitung dieses Übungsblattes (zusammen mit den Lösungen der vorhergehenden Übungsblätter) die gesamte Kette realisiert ist.

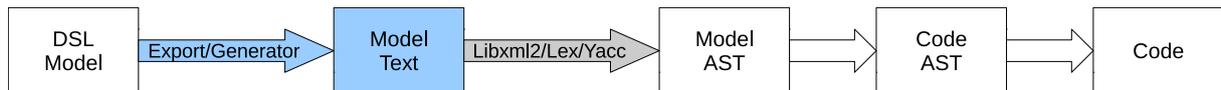


Abbildung 1: Workflow: vom DSL-Modell zum ausführbaren Code

Wie in Übungsblatt 5 bereits festgelegt, sollen als DSL-Modelle Instanzen des von Ihnen zur Modellierung von *Komponentenhierarchien mit hierarchischen Timed Automata* in Übungsblatt 2 entwickelten Metamodells (im Folgenden: HTA-Modelle) verwendet werden. Ein entsprechendes MetaEdit+-Metamodell, welches außerdem einen Generator zur Transformation von HTA-Modellen ins XML-Format bereitstellt, steht zum Download bereit<sup>1</sup>. Nachfolgend wird die verwendete XML-Struktur beispielhaft dargestellt:

```
<tamodel>
  <component name="system">
    <vardecl name="b" type="Integer"/>
    <vardecl name="globalClock" type="Clock"/>
    <component name="environment">
      <vardecl name="x" type="Integer"/>
      <vardecl name="y" type="Integer"/>
      <location name="l0" id="3_175" init="T">
        <location name="l5" id="3_265" init="F">
          <location name="sdf" id="3_295" init="T">
            <exitact lhs="z" rhs="(y+1)*z"/>
          </location>
        </location>
      </location>
      <location name="l2" id="3_235" init="F">
        <transition guard="" tgt="3_265">
          <act lhs="c" rhs="0"/>
        </transition>
      </location>
      <location name="l1" id="3_205" init="T">
        <entryact lhs="a" rhs="4*b"/>
        <entryact lhs="z" rhs="y/2*(4-x)/>
        <doact lhs="b" rhs="x+y*10"/>
        <transition guard="(x & & y & & (z-1 > 2))" tgt="3_265">
        </transition>
        <transition guard="a > b" tgt="3_325">
          <act lhs="a" rhs="b+1"/>
        </transition>
      </location>
    </location>
  </component>
  <component name="system under test">
    ...
  </component>
</tamodel>
```

<sup>1</sup><http://www.informatik.uni-bremen.de/agbs/lehre/ws1011/ses/aufgaben/metamodel.tar.bz2>

**Aufgabe 1:** Implementieren Sie eine Transformation, die aus einer wie oben dargestellten XML-Repräsentation eines HTA-Modells einen Model-AST (s. Übungsblatt 5) instantiiert. Zum Parsen von XML-Dokumenten soll die `libxml2`-Bibliothek verwendet werden. Da in der gegebenen XML-Struktur Expressions lediglich in Form von Strings vorliegen, wird zusätzlich ein Expression-Parser benötigt - ein hierfür geeigneter Parser wird zur Verfügung gestellt<sup>2</sup> und muss nicht implementiert werden.

Alle in der Vorlesung bereits erarbeiteten Teile dieser Aufgabe können wieder von der SeS-Webseite heruntergeladen werden<sup>3</sup>. Die (unvollständige) Implementierung ist geeignet zu ergänzen / anzupassen, wobei auch diesmal die zusätzlichen Hinweise von Übungsblatt 4 zu berücksichtigen sind.

**Aufgabe 2:** Testen Sie Ihre Implementierung der gesamten Kette, indem Sie beispielhaft einige HTA-Modelle erzeugen und auf diese Modelle jeden Transformationsschritt aus Abbildung ?? anwenden, so dass schließlich ausführbarer C-Code produziert wird.

**Abgabe: 15.02.2011 bis 16:00 Uhr**

---

<sup>2</sup><http://www.informatik.uni-bremen.de/agbs/lehre/ws1011/ses/aufgaben/expparser.tar.bz2>

<sup>3</sup><http://www.informatik.uni-bremen.de/agbs/lehre/ws1011/ses/aufgaben/ast.tar.bz2>