

Entwicklung einer domänenspezifischen Sprache (DSL) und eines Compilers

In der Regelungstechnik werden zur Entwicklung von Reglern häufig Blockdiagramme verwendet. Gewisse Problemstellungen lassen sich mit solchen Diagrammen jedoch nur mühsam modellieren, insbesondere wenn dafür sehr viele Blöcke und Verbindungen benötigt werden oder die Ausführungsreihenfolge relevant ist.

Das Ziel dieser Arbeit ist, solche Problemstellungen zu identifizieren und dafür eine textuelle domänenspezifische Sprache (DSL) zu konzipieren, deren Code als Block in die Entwicklungsumgebung des Industriepartners eingebunden werden kann. Für diese Sprache wird ausserdem ein Prototyp von Compiler und Editor entwickelt.

In der Arbeit wird das Konzept der DSL, deren Kern weitgehend zu MATLAB kompatibel ist, vorgestellt. Ihre Ausdrucksstärke erhält die DSL durch eine Kombination von Sprachkonstrukten und semantischen Hervorhebungen durch den Editor. Mit dem Prototyp lassen sich zustandslose Blöcke implementieren, wozu bereits ein grosser Teil des geplanten Sprachkerns verwendet werden kann. Ausserdem werden verschiedene statische Code-Analysen durchgeführt und im Editor Hilfestellungen, wie beispielsweise eine Autovervollständigung, angeboten. Der Prototyp enthält als weiteren Bestandteil eine Standardbibliothek mit einigen grundlegenden Funktionen.

Ein erstes Feedback von Anwendern hat gezeigt, dass die DSL als intuitiv zugänglich wahrgenommen wird und für die häufigsten Problemstellungen bei der Reglerentwicklung eine Lösung bietet.



Diplomand
Silvan Müller

Dozent
Jürgen Spielberger

```
blocks.script
funblock Example
  input x0: double
    name "In0" default 1
  input x1: double
    name "In1" default 0
  output y: double
    name "Out"

  update
    let alpha
    if x0 <= 1e2
      alpha = exp(x0)
    else
      alpha = x0
    end

    y = alpha * fix(x1)
  end
end
```

Beispiel einer Blockimplementierung