

iMOL: Ein experimentelles Werkzeug zur interaktiven Validierung von Softwaremodellen

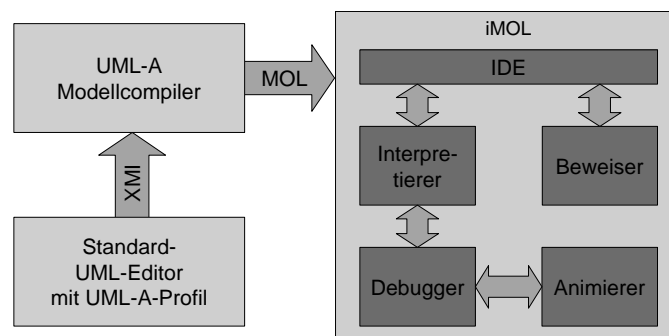
Uwe Thaden, Tobias Buchloh, Friedrich Steimann

Learning Lab Lower Saxony, Universität Hannover
Expo Plaza 1, D-30539 Hannover
{thaden, buchloh, steimann} @learninglab.de

Abstrakt: Ein potentieller Verwendungszweck von Softwaremodellen ist die Validierung der darin festgehaltenen Anforderungen. Wir stellen einen experimentellen Ansatz vor, mit Hilfe dessen sich ein Modell per interaktiver Animation ausführen und auf diese Weise durch den Benutzer validieren läßt.

1 Von UML zu iMOL

Auch die neue Version 2 der Unified Modeling Language (UML) stellt keinen einheitlichen Modellierungsformalismus zur Verfügung, sondern enthält Ungereimtheiten und Unklarheiten; insbesondere fehlt ihr eine präzise Semantik [St04]. Einen ersten Schritt in Richtung auf eine Lösung würde eine Teilsprache mit formaler Semantik darstellen, die sowohl Korrektheitsbeweise (im Sinne einer Modellverifikation) als auch die Modellausführbarkeit (im Sinne einer Modellvalidierung) erlaubt [dC03]. Dies kann erreicht werden, indem man einen UML-Sprachkern (UML-A genannt) isoliert und für diesen eine Abbildung in eine Sprache findet, die die geforderten Eigenschaften besitzt. Eine solche Sprache, die sog. Modeling Object Language (MOL), haben wir vorgeschlagen [SV04] und ein Werkzeug mit dem Namen iMOL dazu entwickelt, das MOL-Programme schrittweise und interaktiv auszuführen erlaubt und dabei eine Animation des Ausgangsmodells triggert [Bu03]. Die Architektur von iMOL und seine Einbettung in den Modellierungsprozeß sind in untenstehender Abbildung skizziert.



2 Sinn und Nutzen von iMOL

iMOL ist ein experimentelles System, das mehreren Zwecken dient:

1. Es soll die Minimalanforderungen an die Ausdrucksstärke einer Modellierungssprache mit ausführbarer Semantik (hier: MOL) zu explorieren erlauben.
2. Es soll die praktischen Anforderungen an eine Modellvalidierung auf Basis von Modellanimationen im Stile von [ST+02] zu untersuchen erlauben. Daraus sollen Rückschlüsse auf notwendige Eigenschaften der Modellierungssprache (hier: UML-A) gezogen werden (wie müssen Diagramme in Beziehung stehen, damit sie zusammen eine ausführbare Systemspezifikation ergeben? Wie sind Einstiegspunkte (Menüs) zu modellieren?).
3. Es soll der Untersuchung dienen, inwieweit sich die Ausführbarkeit (im Sinne einer Modellanimation) mit der Beweisbarkeit bestimmter Modelleigenschaften unter einem einzigen Formalismus (hier: MOL) vereinen lässt.

3 Status Quo und nächste Schritte

Die erste Iteration der (parallelen) Entwicklung von UML-A und MOL gemeinsam mit ihren Werkzeugen, iMOL und dem Modellcompiler MOCO [Ha03], ist abgeschlossen. Die Übersetzung von UML nach MOL ermöglicht es, Softwaremodelle in einer Sprache mit operationaler Semantik auszudrücken. Unser experimenteller Ansatz erlaubt es uns, Sprache und Werkzeuge parallel zu entwickeln und in der Praxis zu erproben.

Projektseite: <http://www.kbs.uni-hannover.de/Projekte/iMOL>

Referenzen

- [Bu03] Buchloh, T.: Entwurf und Implementierung eines Interpretierers für die Sprache MOL (model object language). Bachelorarbeit Institut für Informationssysteme, Universität Hannover (2003).
- [dC03] de Champeaux, D.: Forum: Extending and shrinking UML. *Communications of the ACM* 46:3 (2003), S. 11–12.
- [Ha03] Hagemann, P.: UML-Diagramme und ihre Übersetzung in die Modell-Objekt-Sprache MOL. Diplomarbeit Institut für Informationssysteme, Universität Hannover (2003).
- [St04] Steimann, F.: UML-A oder warum die Wissenschaft ihre eigene einheitliche Modellierungssprache haben sollte. *Modellierung 2004* (in diesem Band).
- [SV04] Steimann, F.; Vollmer, H.: Exploiting the practical limitations of diagrammatic specifications for model validation and execution. In Vorbereitung für: *Journal on Software & System Modeling*.
- [ST+02] Steimann, F.; Thaden, U.; Siberski, W.; Nejd, W.: Animiertes UML als Medium für die Didaktik der objektorientierten Programmierung. *Modellierung 2002; GI Lecture Notes in Informatics* (2002).