

# Spezifikation von Echtzeiteigenschaften mit OCL für eine MFERT-basierte Fallstudie

Stephan Flake, Wolfgang Müller

C-LAB, Universität Paderborn, Fürstenallee 11, 33102 Paderborn

email: {flake, wolfgang}@c-lab.de

## Zusammenfassung

Die Object Constraint Language (OCL) wurde entwickelt, um Modellierer bei der Spezifikation von Modelleigenschaften im Kontext gegebener UML Diagramme zu unterstützen [3]. Jedoch ist es leider nicht möglich, Eigenschaften, die das dynamische Verhalten betreffen, mit der derzeitigen Version von OCL auszudrücken.

Wir präsentieren eine Erweiterung von OCL Version 1.4 zur Spezifikation von zustandsbasierten, zeitbehafteten Eigenschaften. Obwohl OCL und unsere Erweiterung in erster Linie im Rahmen der UML entwickelt worden sind, ist es auch möglich, mit dieser Sprache Eigenschaften für andere objektbasierte Formalismen zu beschreiben. Wir veranschaulichen dies anhand einer Anwendung bei der Modellprüfungsspezifikation in Verbindung mit MFERT, einer praxisrelevanten Notation, die im Projekt GRASP zur Modellierung von Produktionsabläufen eingesetzt wird.

**Einleitung.** In den zunehmend komplexeren Modellen von Produktionsabläufen ist selbst für erfahrene Ingenieure nicht mehr nachvollziehbar, ob mit einem vorliegenden Modell eine korrekte Ausführung gemäß der Anforderungsspezifikation gewährleistet ist. Auch Simulationsdurchläufe reichen nicht mehr aus, um alle kritischen Situationen aufzudecken, weil eine umfassende Evaluierung zu viel Zeit in Anspruch nehmen würde.

Im Gegensatz zur Simulation wird bei der sog. Modellprüfung der *gesamte* Modellzustandsraum untersucht. Als ein wesentlicher Grund für die mangelnde Akzeptanz dieser Methode im praktischen Einsatz wird oft angeführt, dass zusätzlich zum eigentlichen Modell die einzuhaltenden Eigenschaften als temporallogische Formeln spezifiziert werden müssen, die als schwer verständlich angesehen werden. Verschiedene Vorschläge sind bereits unterbreitet worden, um die Formulierung von Eigenschaften zu vereinfachen, z.B. Kataloge wiederkehrender Spezifikationsmuster oder graphische Ansätze, mit denen bestimmte Ausführungssequenzen dargestellt werden können. Von diesen Ansätzen konnte sich jedoch bisher keiner in der Praxis durchsetzen. Auf der anderen Seite hat die OCL als fester Bestandteil der UML zur Spezifikation von Modelleigenschaften einen signifikanten Bekanntheitsgrad erreicht.

**Zeitorientierte Erweiterung von OCL.** OCL wird hauptsächlich in Klassendiagrammen eingesetzt, um Invarianten sowie Vor- und Nachbedingungen von Operationen zu formulieren. Obwohl OCL auch in verhaltensbeschreibenden Diagrammen eingesetzt werden kann, z.B. in Statecharts für Bedingungen an Zustandsübergängen, ist es zurzeit nicht möglich, dynamische oder zeitbehaftete Verhaltenseigenschaften mit OCL auszudrücken.

Wir haben die OCL so erweitert, dass auch zustandsbasierte, zeitbehaftete Modelleigenschaften formuliert werden können. Im Gegensatz zu anderen temporalen OCL-Erweiterungen ist unser Ansatz in ein Metamodell [1] integriert, so wie es auch für die anderen Teile der UML üblich ist. Dies hat auch zur Folge, dass sich die resultierende Syntax mit der offiziellen OCL-Notation deckt.

Bisher haben wir unsere OCL-Erweiterung in Verbindung mit UML Statecharts vorgestellt [2], veranschaulichen hier aber auch die Anwendung auf UML Aktivitätendiagramme und MFERT Ablaufdiagramme. Die Beschreibungsmethode MFERT (Modell der FERTigung) basiert u.a. auf einer Petrinetz-ähnlichen Ausführungssemantik, die zwischen Fertigungsvorgängen (FV) und Fertigungselementen (FE) unterscheidet. Während FE-Klassen verfügbare Fertigungselemente (Material und Ressourcen) repräsentieren, werden Fertigungsvorgänge regelbasiert durch endliche Automaten mit zeitbehafteten Transitionen angegeben.

Wir haben für unsere OCL-Erweiterung eine Abbildung in CCTL – eine zeitbehaftete Variante der Temporallogik CTL – definiert, um so eine automatische Übersetzung in temporallogische Formeln zu ermöglichen. Zusammen mit einer Abbildung von MFERT in die Eingabesprache RIL des Modellprüfers RAVEN [4] kann auf diese Weise eine automatische Modellverifikation stattfinden.

**Ziel.** Dadurch, dass wir auf der bereits verbreiteten Sprache OCL aufsetzen, erwarten wir von unserer Erweiterung eine im praktischen Einsatz verwendbare Spezifikationsprache, mit der auch komplexe, zeitbehaftete Modelleigenschaften formuliert werden können, ohne dass man sich direkt mit Temporallogik auseinander setzen muss. Dabei beschränken wir uns nicht auf UML Diagramme, sondern zeigen mit der Anwendung auf MFERT, dass auch andere objektbasierte Formalismen von diesem Ansatz profitieren können.

## Literatur

- [1] T. Baar, R. Hähnle. An Integrated Metamodel for OCL Types. In: *Proc. of OOPSLA 2000, Workshop Refactoring the UML: In Search of the Core*, 2000.
- [2] S. Flake, W. Müller. An OCL Extension for Real-Time Constraints. In: T. Clark and J. Warmer (eds.), *Advances in Object Modelling with the OCL*, Oktober 2001, Springer-Verlag, Heidelberg. (zur Veröffentlichung akzeptiert)
- [3] Object Management Group (OMG). *UML Unified Modeling Language Specification, Version 1.3, Chapter 7: Object Constraint Language*, März 2000.
- [4] J. Ruf. RAVEN: Real-Time Analyzing and Verification Environment. In: *Journal on Universal Computer Science (J.UCS)*, Februar 2001, Springer-Verlag, Heidelberg.