From Core OCL Invariants to Nested Graph Constraints

Thorsten Arendt\textsuperscript{1}, Annegret Habel\textsuperscript{2}, Hendrik Radke\textsuperscript{2}, and Gabriele Taentzer\textsuperscript{1}

\textsuperscript{1} Philipps-Universität Marburg, Germany
\{arendt,taentzer\}@informatik.uni-marburg.de
\textsuperscript{2} Universität Oldenburg, Germany
\{habel,radke\}@informatik.uni-oldenburg.de

Abstract. Meta-modeling including the use of the Object Constraint Language (OCL) forms a well-established approach to design domain-specific modeling languages. This approach is purely declarative in the sense that instance construction is not needed and not considered. In contrast, graph grammars allow the stepwise construction of instances by the application of transformation rules. In this paper, we consider meta-models with Core OCL invariants and translate them to nested graph constraints for typed attributed graphs. Models and meta-models are translated to instance and type graphs. We show that a model satisfies a Core OCL invariant iff its corresponding instance graph satisfies the corresponding nested graph constraint. The aim of this work is to establish a first formal relation between meta-modeling and the theory of graph transformation including constraints to come up with an integrated approach for defining modeling languages in an optimal way in the future.

Keywords: Meta modeling, OCL, graph constraints, application conditions.

1 Introduction

The trend towards model-based and model-driven software development causes a need of new, mostly domain-specific modeling languages with well-designed tool support. Therefore we need methods and techniques to define modeling languages and their tooling precisely and also intuitively. A comprehensive language definition needs the declarative as well as the constructive paradigm to specify language properties, to construct and recognize language instances as well as to modify them. Nowadays, modeling languages are typically defined by meta-models following purely the declarative approach. In this approach, language properties are specified by the Object Constraint Language (OCL) \cite{1}.

* This work is partly supported by the German Research Foundation (DFG), Grant HA 2936/4-1 (Meta modeling and graph grammars: integration of two paradigms for the definition of visual modeling languages).
In contrast, graph grammars have shown to be suitable and natural to specify visual languages in a constructive way, by using graph transformation [2]. Recently, nested graph constraints [3] have been developed to include also the declarative element into graph grammars. To ensure that a graph grammar fulfills a set of graph constraints, they can be translated to application conditions of graph rules such that all graphs fulfilling the constraints in the beginning keep on fulfilling them after applying graph rules being extended by translated application conditions.

While typed attributed graphs form an adequate formalization of instance models that are typed over a meta-model [4], the relation of OCL constraints to nested graph constraints has not been considered yet. We are interested in investigating this relation, since the translation of graph constraints to application conditions for rules opens up a way to combine declarative and constructive elements in a formal approach. By translating OCL to nested graph constraints, such an integration of declarative and constructive elements becomes possible also in the meta-modeling approach. It shall open up a way to translate OCL constraints to application conditions of model transformation rules making applications as e.g. auto-completion of model editing operations to consistent models possible.

As a basis, models and meta-models (without OCL constraints) are translated to instance and type graphs. In this paper, we investigate the relation of meta-models including OCL constraints and nested graph constraints for typed attributed graphs. It turns out that Core OCL invariants [5], i.e. Boolean expressions over navigations based on the type system, can be well translated to nested graph constraints. The aim of this work is to establish a first formal relation between meta-modeling and the theory of graph transformation to come up with an integrated approach for defining modeling languages in an optimal way in the future.

This paper is structured as follows: The next section presents OCL in a nutshell focusing on Core OCL invariants. Section 3 shows typed attributed graphs and graph morphisms as well as nested graph conditions. Section 4 presents our main contribution of this paper, the translation of Core OCL invariants to nested graph constraints. Section 5 discusses how Core OCL invariants can be translated to equivalent application conditions of graph rules. Section 6 compares to related work and concludes the paper.

## 2 Core OCL Invariants

In this section, we recall Core OCL constraints presenting a small example first and formally defining their syntax and semantics thereafter, according to the work by Richters [6] that went into the OCL specification by the OMG [1]. For illustration purposes, we use the following meta-model for simple Petri nets to recall OCL.

**Example 1.** A Petri net (PetriNet) is composed of places (Place) or transitions (Transition) which are linked together by arcs (ArcTP for linking exactly one