Software Design, Modeling and Analysis in UML
Lecture 03: Object Constraint Language (OCL)

2013-10-28
Prof. Dr. Andreas Podelski, Dr. Bernd Westphal
Albert-Ludwigs-Universität Freiburg, Germany

Contents & Goals

Last Lecture:
• Basic Object System Signature / CB
• System State \( \sigma \in \Sigma \) / BW
(Smells like they're related to class/object diagrams, officially we don't know yet...)

This Lecture:
• Educational Objectives:
  • Please explain this OCL constraint.
  • Please formalize this constraint in OCL.
  • Does this OCL constraint hold in this system state?
  • Can you think of a system state satisfying this constraint?
  • Please un-abbreviate all abbreviations in this OCL expression.
  • In what sense is OCL a three-valued logic? For what purpose?
• Content:
  • OCL Syntax, OCL Semantics over system states

What is OCL? And What is It Good For?

Most prominent:
• Write down requirements supposed to be satisfied by all system states.
  Oftentimes targeting all alive objects of a certain class.
• Not unknown:
  • Write down pre/post-conditions of methods (Behavioural Features).
  Then evaluated over two system states.

What's It Good For?

• Most prominent:
  • Write down requirements supposed to be satisfied by all system states.
  Oftentimes targeting all alive objects of a certain class.
• Not unknown:
  • Write down pre/post-conditions of methods (Behavioural Features).
  Then evaluated over two system states.

OCL/Beispiel

TeamMember
name : String
age : Integer

participants 2..* meet meetings
* title : String
numParticipants : Integer
start : Date
duration: Time

Meeting
move(newStart : Date)

context TeamMember
inv:
  age => 18

context Meeting
inv:
  duration > 0

(\text{C}) Prof. Dr. P. Thiemann, http://proglang.informatik.uni-freiburg.de/teaching/swt/2008/
Given an OCL expression \( \phi \), the evaluated set \( \{ \tau | \phi \} \) represents the set of all \( \tau \) for which \( \phi \) is satisfied.

When \( \phi \) is satisfied by all system states, \( \{ \tau | \phi \} \) is a certain class. Oftentimes, targeting all alive objects of a certain class, a valuation of \( \Sigma \in \{ \tau | \phi \} \) can be considered.

Thenevaluatedover systemstates. Two

Thenevaluatedover systemstates.
Iterate: Intuitive Semantics (Formally: later)

forexample

\begin{itemize}
\item \(\tau\) is an expression of type \(\omega_1, \ldots, \omega_n\).
\item \(\tau\) gives the initial value \(r\) of the result \(\tau\).
\item \(\tau\) is a collection type \(\tau\), i.e., if \(\tau\) is an object type \(\tau\), then
\item \(\tau\) is a collection type \(\tau\), i.e., if \(\tau\) is an object type \(\tau\), then
\end{itemize}

OCL Syntax/Notational Conventions for Expressions
More Notational Conventions

- For context self: τ

  \[
  \text{inv} : \text{expr}
  \]

  we may alternatively write (abbreviate as)

  \[
  \text{context } \tau \text{ inv : expr}
  \]

  within the latter abbreviation, we may omit the "self" in expr, i.e. for self. v and self. r we may alternatively write (abbreviate as)

  v and r

Examples (from lecture "Softwaretechnik 2008")

TeamMember

name : String
age : Integer

Location

participants : 2..*meetings

*title : String
numParticipants : Integer
start : Date
duration : Time

Meeting

move(newStart : Date)

1

\[
\text{context Meeting} \text{ inv : participants} \rightarrow \text{iterate (i : TeamMember; n : Integer = 0 | n + i.age)}
\]

\[
\text{participants} \rightarrow \text{size} > 25
\]

\[
\text{context Location} \text{ inv : name = "Lobby" implies meeting} \rightarrow \text{isEmpty()}
\]

(C) Prof. Dr. P. Thiemann, http://proglang.informatik.uni-fr

Among others:

- Enumeration types
- Type hierarchy
- Complete list of arithmetical operators
- The two other collection types Bag and Sequence
- Casting
- Run-time type information
- Pre/postconditions (may be later, when we officially know what an operation is)
- ...

References
References


