

Analyzing Semantic Properties of OCL Operations by Uncovering Interoperational Relationships

Mirco Kuhlmann and Martin Gogolla

Database Systems Group
Department for Mathematics and Computer Science
University of Bremen

September 30, 2007



Motivation

Objective

- Compile Semantic Relationships (Equivalences)
- Initiate Discussions
- Develop an OCL Benchmark (← large Variety of OCL Engines)

Subject

- OCL Collection Operations
- Logic related: **exists**, **forAll**, **one**
- Database related: **reject**, **select**
- Functional Programming related: **collect**

Property of one

$C \rightarrow \text{one}(\dots | e) \equiv$

$C \rightarrow \text{exists}(\dots | e)$ and

$C \rightarrow \text{forAll}(x, y | e_x \text{ and } e_y \text{ implies } x = y)$

Property of one

$C \rightarrow \text{one}(\dots | e) \equiv$

$C \rightarrow \text{exists}(\dots | e)$ and

$C \rightarrow \text{forAll}(x, y | e_x \text{ and } e_y \text{ implies } x = y)$

$\text{Set}\{1, 1, 2, 3\} \rightarrow \text{one}(x \mid x < 2) \equiv \text{true}$

$\text{Set}\{1, 1, 2, 3\} \rightarrow \text{exists}(x \mid x < 2)$ and

$\text{Set}\{1, 1, 2, 3\} \rightarrow \text{forAll}(x, y \mid$

$(x < 2) \text{ and } (y < 2) \text{ implies } x = y) \equiv \text{true}$

Valid for Bag{1, 1, 2, 3}?

Property of one

$C \rightarrow \text{one}(\dots | e) \equiv$
 $C \rightarrow \text{exists}(\dots | e)$ and
 $C \rightarrow \text{forAll}(x, y | e_x \text{ and } e_y \text{ implies } x = y)$

$\text{Set}\{1, 1, 2, 3\} \rightarrow \text{one}(x \mid x < 2) \equiv \text{true}$

$\text{Set}\{1, 1, 2, 3\} \rightarrow \text{exists}(x \mid x < 2)$ and

$\text{Set}\{1, 1, 2, 3\} \rightarrow \text{forAll}(x, y \mid$
 $(x < 2) \text{ and } (y < 2) \text{ implies } x = y) \equiv \text{true}$

Valid for Bag{1, 1, 2, 3}?

$\text{Bag}\{1, 1, 2, 3\} \rightarrow \text{one}(x \mid x < 2) \equiv \text{false}$

$\text{Bag}\{1, 1, 2, 3\} \rightarrow \text{exists}(x \mid x < 2)$ and

$\text{Bag}\{1, 1, 2, 3\} \rightarrow \text{forAll}(x, y \mid$
 $(x < 2) \text{ and } (y < 2) \text{ implies } x = y) \equiv \text{true}$

Relationship Compilation (1)

Kind	Left	Right
Logic	$C \rightarrow \text{exists}(\dots e)$	$\text{not } C \rightarrow \text{forAll}(\dots \text{not } e)$
	$C \rightarrow \text{forAll}(\dots e)$	$\text{not } C \rightarrow \text{exists}(\dots \text{not } e)$
Set	$C \rightarrow \text{one}(\dots e)$	$C \rightarrow \text{exists}(\dots e)$ and $C \rightarrow \text{forAll}(x, y e_x \text{ and } e_y \text{ implies } x = y)$
Set	$C \rightarrow \text{one}(\dots e)$	$C \rightarrow \text{exists}(x e_x \text{ and } C \rightarrow \text{forAll}(y e_y \text{ implies } x = y))$
DB related		
Inter-disciplinary		
Collect		

Relationship Compilation (1)

Kind	Left	Right
Logic	$C \rightarrow \text{exists}(\dots e)$ $C \rightarrow \text{forall}(\dots e)$	$\text{not } C \rightarrow \text{forall}(\dots \text{not } e)$ $\text{not } C \rightarrow \text{exists}(\dots \text{not } e)$
Set	$C \rightarrow \text{one}(\dots e)$	$C \rightarrow \text{exists}(\dots e)$ and $C \rightarrow \text{forall}(x, y e_x \text{ and } e_y \text{ implies } x = y)$
Set	$C \rightarrow \text{one}(\dots e)$	$C \rightarrow \text{exists}(x e_x \text{ and } C \rightarrow \text{forall}(y e_y \text{ implies } x = y))$
DB related	$C \rightarrow \text{reject}(\dots e)$ $C \rightarrow \text{select}(\dots e)$	$C \rightarrow \text{select}(\dots \text{not } e)$ $C \rightarrow \text{reject}(\dots \text{not } e)$
Inter-disciplinary	$C \rightarrow \text{exists}(\dots e)$ $C \rightarrow \text{exists}(\dots e)$ $C \rightarrow \text{forall}(\dots e)$ $C \rightarrow \text{forall}(\dots e)$ $C \rightarrow \text{one}(\dots e)$ $C \rightarrow \text{one}(\dots e)$	$C \rightarrow \text{reject}(\dots e) \rightarrow \text{size}() < C \rightarrow \text{size}()$ $C \rightarrow \text{select}(\dots e) \rightarrow \text{notEmpty}()$ $C \rightarrow \text{reject}(\dots e) \rightarrow \text{isEmpty}()$ $C \rightarrow \text{select}(\dots e) = C$ $C \rightarrow \text{reject}(\dots e) \rightarrow \text{size}() = C \rightarrow \text{size}() - 1$ $C \rightarrow \text{select}(\dots e) \rightarrow \text{size}() = 1$
Collect	$C \rightarrow \text{exists}(\dots e)$ $C \rightarrow \text{exists}(\dots e)$ $C \rightarrow \text{forall}(\dots e)$ $C \rightarrow \text{forall}(\dots e)$ $C \rightarrow \text{one}(\dots e)$	$C \rightarrow \text{collect}(\dots e) \rightarrow \text{includes}(\text{true})$ $C \rightarrow \text{collect}(\dots e) \rightarrow \text{asSet}() \rightarrow \text{one}(e e)$ $C \rightarrow \text{collect}(\dots e) \rightarrow \text{excludes}(\text{false})$ $\text{let } s = C \rightarrow \text{collect}(\dots e) \rightarrow \text{asSet}() \text{ in}$ $C \rightarrow \text{notEmpty}() \text{ implies } s \rightarrow \text{one}(\text{true}) \text{ and } s \rightarrow \text{one}(e e)$ $C \rightarrow \text{collect}(\dots e) \rightarrow \text{count}(\text{true}) = 1$

Relationship Compilation (2)

Operation	Iterate Expression
exists	$C \rightarrow \text{exists}(\dots e)$
	$C \rightarrow \text{iterate}(\dots ; r : \text{Boolean} = \text{false} r \text{ or } e)$
forall	$C \rightarrow \text{forall}(\dots e)$
one	$C \rightarrow \text{one}(\dots e)$
reject	$C \rightarrow \text{reject}(\dots e)$
select	$C \rightarrow \text{select}(\dots e)$
collect <i>Set, Bag</i> <i>Sequence</i> <i>Sequence</i>	$C \rightarrow \text{collect}(\dots e)$

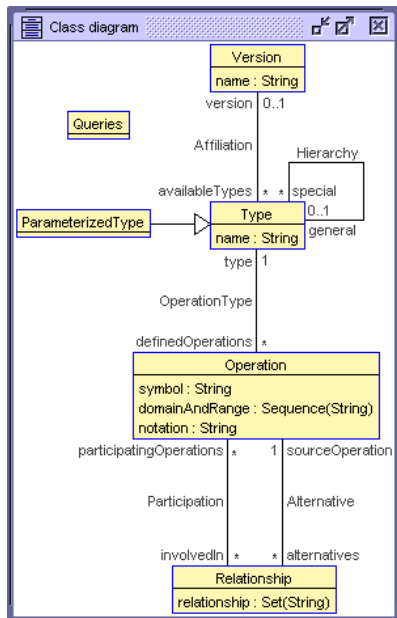
Relationship Compilation (2)

Operation	Iterate Expression
exists	<code>C->exists (... e)</code>
	<code>C->iterate (... ; r:Boolean = false r or e)</code>
forAll	<code>C->forAll (... e)</code>
	<code>C->iterate (... ; r:Boolean = true r and e)</code>
one	<code>C->one (... e)</code>
	<code>C->iterate (... ; r:Sequence (Boolean) = Sequence {false, false} if r->first () then Sequence {true, false} else Sequence {r->last () and e, (r->last () and e) xor (r->last () or e) } endif)->last ()</code>
reject	<code>C->reject (... e)</code>
	<code>C->iterate (elem; r:ct = oclEmpty (ct) if e then r else r->including (elem) endif)</code>
select	<code>C->select (... e)</code>
	<code>C->iterate (elem; r:ct = oclEmpty (ct) if e then r->including (elem) else r endif)</code>
collect	<code>C->collect (... e)</code>
	<i>Set, Bag</i> <code>C->iterate (... ; r:Bag (et) = oclEmpty (Bag (et)) r->including (e)</code>
	<i>Sequence</i> <code>C->iterate (... ; r:Sequence (et) = oclEmpty (Sequence (et)) r->append (e)</code> (also possible <code>r->including (e)</code>)
	<i>Sequence</i> <code>C->iterate (... ; r:Sequence (et) = oclEmpty (Sequence (et)) Sequence {r, Sequence {e}}->flatten ())</code>

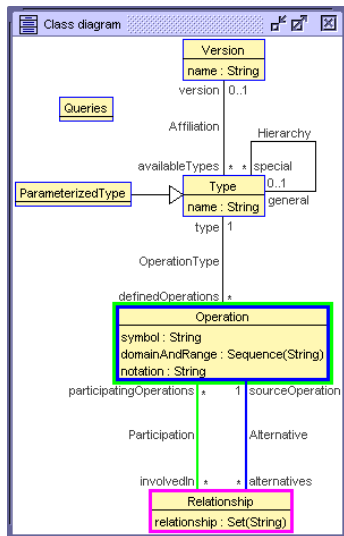
Relationship Warehouse

- Record the Equivalences
- Allow various Queries for different Purposes
- Implementation
 - ▶ Tool: USE (UML-based Specification Environment)
 - ▶ Model: UML Class Diagram
 - ▶ Content: UML Object Diagram
 - ▶ Queries: OCL Expressions

Warehouse Model



Querying the Warehouse



```
q.getRelationshipBySymbolAndParticipating(  
  'one', 'Mark Richters (USE)',  
  'Collection', Set{'select'})
```

```
Set{  
  'X->one(elem : ElemType | Expr) ==  
  X->select(elem : ElemType | Expr)  
  ->size() = 1'  
} : Set(String)
```

Conclusion and Future Work

- Semantic Analysis of OCL Relationships
 - ▶ Proposing Equivalences
 - ▶ Uncovering Assumptions for Equivalences → Modifications of Equivalences
- OCL Benchmark
 - ▶ Many Tools (→ Quality Assurance)
 - ▶ Making General Equivalences Applicable through Concrete Examples
 - ▶ Measuring Execution Times
 - ★ Equivalences
 - ★ Special OCL Terms
(e.g. `Set{1..1000}` → `select ('Square Numbers')`)

Property of one

$C \rightarrow \text{one}(\dots | e) \equiv$
 $C \rightarrow \text{exists}(\dots | e)$ and
 $C \rightarrow \text{forAll}(x, y | e_x \text{ and } e_y \text{ implies } x = y)$

$\text{Set}\{1, 1, 2, 3\} \rightarrow \text{one}(x \mid x < 2) \equiv \text{true}$

$\text{Set}\{1, 1, 2, 3\} \rightarrow \text{exists}(x \mid x < 2)$ and

$\text{Set}\{1, 1, 2, 3\} \rightarrow \text{forAll}(x, y \mid$
 $(x < 2) \text{ and } (y < 2) \text{ implies } x = y) \equiv \text{true}$

Valid for Bag{1, 1, 2, 3}?

$\text{Bag}\{1, 1, 2, 3\} \rightarrow \text{one}(x \mid x < 2) \equiv \text{false}$

$\text{Bag}\{1, 1, 2, 3\} \rightarrow \text{exists}(x \mid x < 2)$ and

$\text{Bag}\{1, 1, 2, 3\} \rightarrow \text{forAll}(x, y \mid$
 $(x < 2) \text{ and } (y < 2) \text{ implies } x = y) \equiv \text{true}$

Conclusion and Future Work

- Semantic Analysis of OCL Relationships
 - ▶ Proposing Equivalences
 - ▶ Uncovering Assumptions for Equivalences → Modifications of Equivalences
- OCL Benchmark
 - ▶ Many Tools (→ Quality Assurance)
 - ▶ Making General Equivalences Applicable through Concrete Examples
 - ▶ Measuring Execution Times
 - ★ Equivalences
 - ★ Special OCL Terms
(e.g. `Set{1..1000}` → `select ('Square Numbers')`)

Thank you for your attention!