The Dresden OCL Toolkit and the Business Rules Approach

Birgit Demuth

Technische Universität Dresden

Fakultät Informatik

Contents

- Introduction
- Architecture of the Dresden OCL Toolkit
  - OCL1.3 based
  - OCL2.0 based
- OCL in Business Rules Solutions
- Lessons Learned and Conclusions

Introduction

OCL at a Glance

- Object Constraint Language (OCL)
- MOF related OMG standard
- Part of the Unified Modeling Language (UML)
- Language for defining constraints in different UML diagrams
  - Declarative
  - Formal
  - Side-effects-free
- OCL adds precision to the mostly graphical/textual specifications of software projects
- Growing acceptance by more running OCL tools in last years

The EBRC Question

What kind of support

can OCL provide

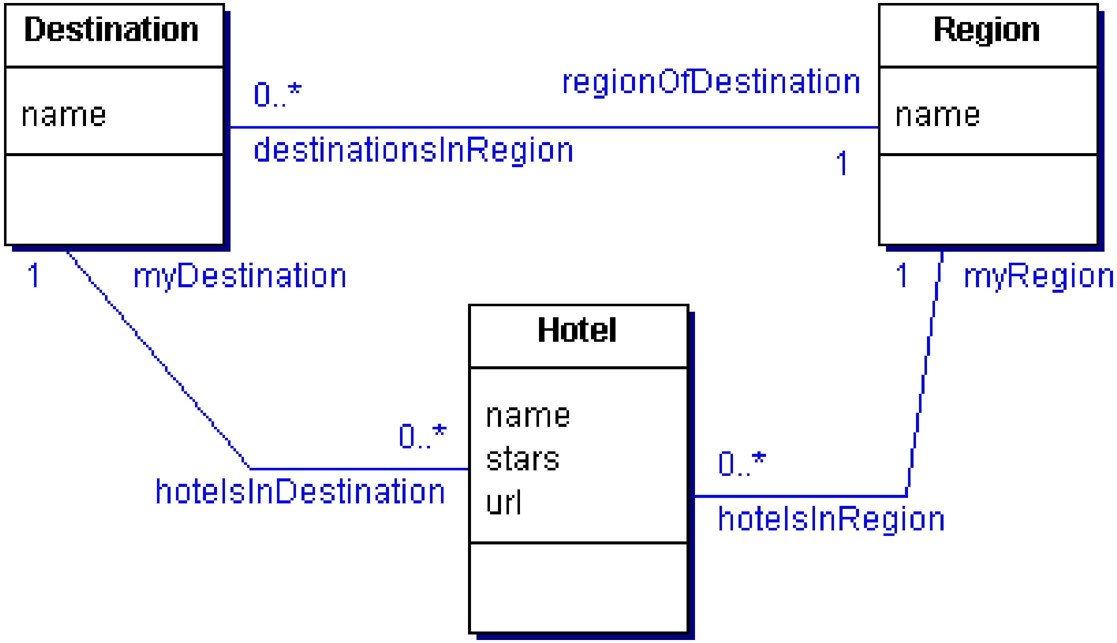
for the specification and evaluation of business rules?

(Software engineering perspective)

Typical OCL Use Cases

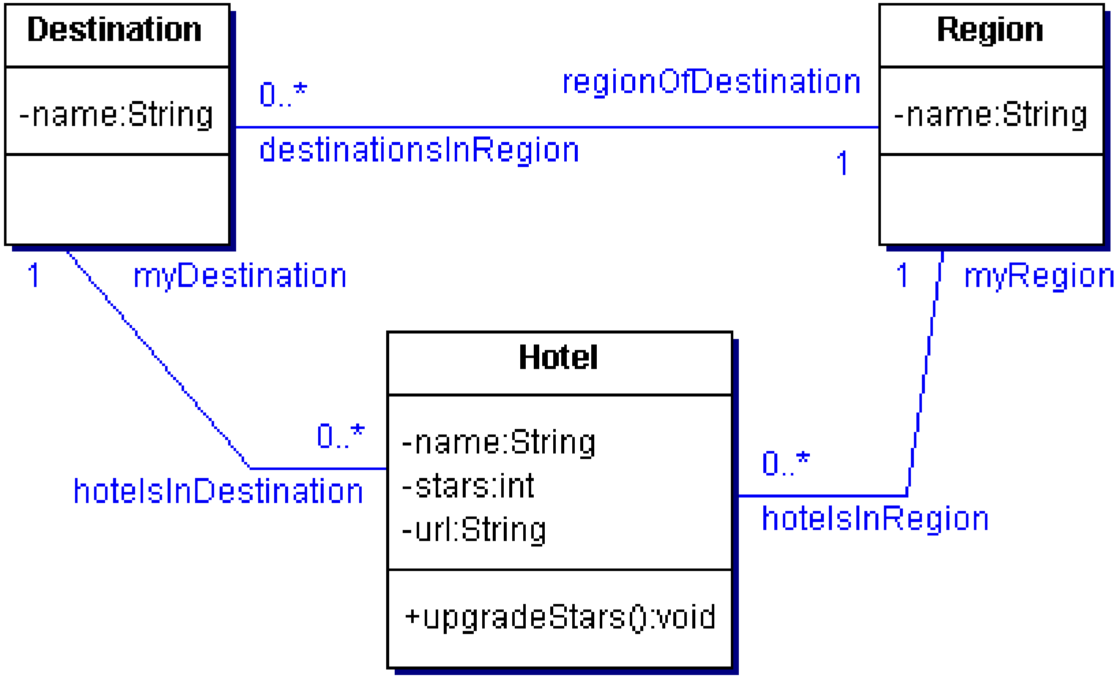
- Development of business applications
  - Specification and evaluation of business rules
  - Specification and evaluation of modeling guidelines/„profiles“
- Development of CASE tools
  - Checking the consistency of software models according to the UML metamodel
  - Providing OCL support
- Model-driven software engineering/MDA
- Specification of test cases and generation of test code

OCL by a Business Rule Example (Invariant)



```
context Destination inv iRegion:
hotelsInDestination->forAll
(myRegion=self.regionOfDestination)
```

OCL by a Business Rule Example (Pre/Post Condition)



```
context Hotel::upgradeStars()
pre: stars>=0
post: stars = stars@pre + 1
```

OCL 2.0
as a Query Language

- Specification of query operations in the UML model, e.g. the names of all hotels in a region

```
context Region::getNamesOfHotels:Bag(name)
body: hotelsInRegion->collect(name)
```

OCL Expression

- Further new use cases of OCL2.0 expressions:
  - Definition of reusable OCL expressions (attributes/ operations) (`def`)
  - Derivation rules (`derive`) for attributes/association ends
  - Specification of initial values for attribute/ass. ends (`init`)

Examples for OCL Constraints at the M2 Layer

- Ensuring the consistency of software models
  - Well-Formedness Rules (WFRs) in the UML metamodel, e.g. circular inheritance is not allowed:

```
context GeneralizableElement:
not self.allParents->includes(self)
```

- UML Modeling guidelines/Profiles
  - Java specific, e.g. there should no multiple inheritance:

```
context GeneralizableElement inv MR1:
self.generalization->size<= 1
```

Architecture of the Dresden OCL Toolkit

OCL1.3 based
OCL2.0 based

Dresden OCL Toolkit

- Dresden OCL Toolkit available as Open Source:
  - http://dresden-ocl.sourceforge.net/
- Modular architecture with cleanly defined interfaces
- Java-based
- Intention is the reuse in (mostly UML CASE) tools that are need in some manner the specification/evaluation of OCL constraints

Dresden OCL Toolkit Available Tools (OCL1.3 based)

- OCLCore (editing, parsing, type checking, normalization)
- OCL2Java (Java code generation)
- OCLInjector4Java (Java code instrumentation)
- OCL2SQL (SQL code generation)
- OCLInterpreter (dynamic OCL constraint evaluation)

Integration of OCL into a UML Tool

- OCL by oneself is nothing!
- Constraints only live together with a UML model!
- Two ways of integration:
  - **Tight** integration

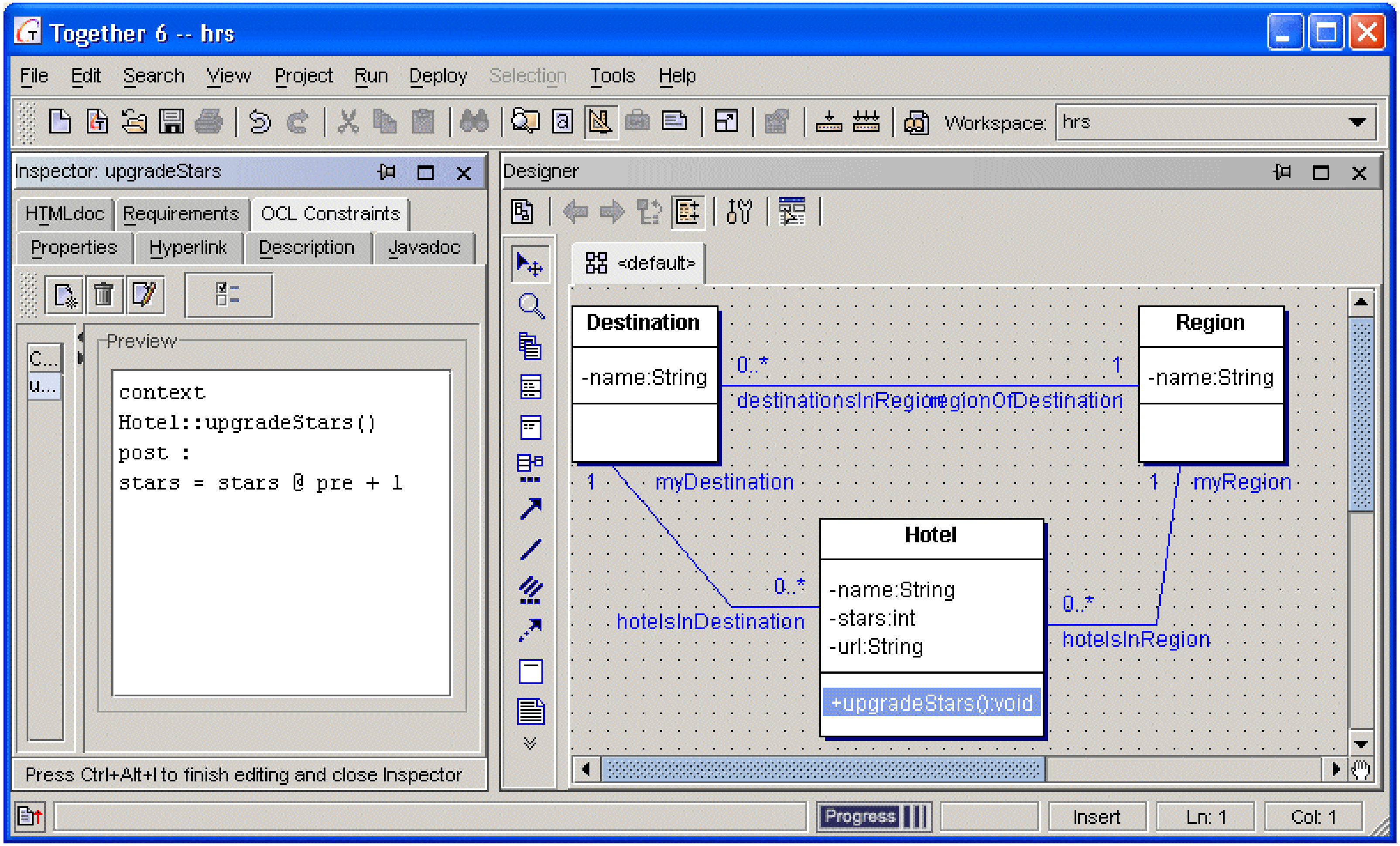
    → OCL tool as an add-in of the UML CASE tool
  - **Loose** integration by XMI

    → XMI file as „Repository“ of the UML model

Existing Integrations of Dresden OCL Tools

- ArgoUML
- Poseidon
- MetaBoss
- Together Control Center
- Rational Rose

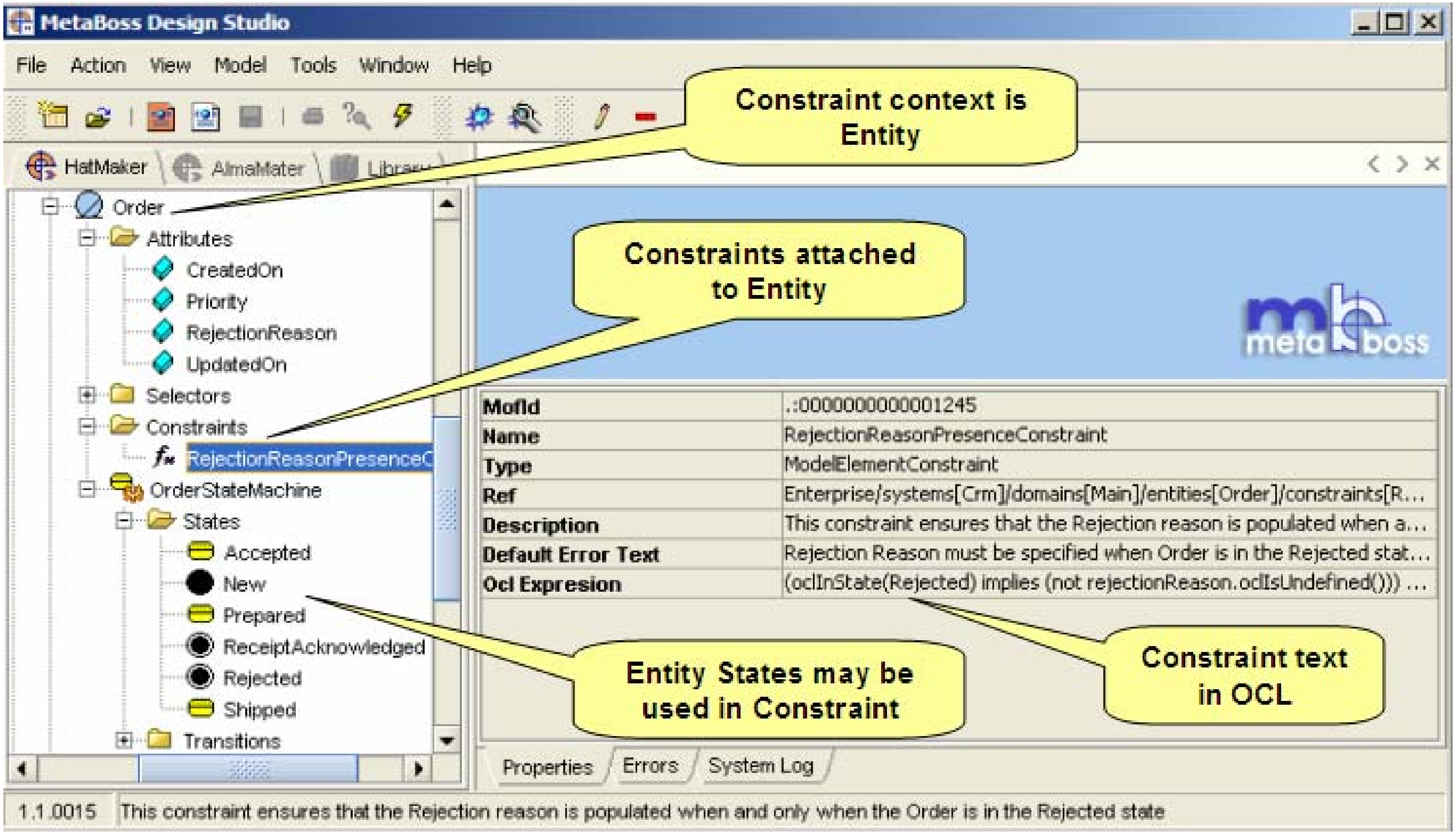
OCLCore
integrated into Together



```
context
Hotel::upgradeStars()
post :
stars = stars @ pre + 1
```

Press Ctrl+Alt+I to finish editing and close Inspector

MetaBoss



MofId	.:0000000000001245
Name	RejectionReasonPresenceConstraint
Type	ModelElementConstraint
Ref	Enterprise/systems[Crm]/domains[Main]/entities[Order]/constraints[R...
Description	This constraint ensures that the Rejection reason is populated when a...
Default Error Text	Rejection Reason must be specified when Order is in the Rejected stat...
Ocl Expresion	(oclInState(Rejected) implies (not rejectionReason.oclIsUndefined())) ...

1.1.0015 This constraint ensures that the Rejection reason is populated when and only when the Order is in the Rejected state

Constraint context is Entity

Constraints attached to Entity

Entity States may be used in Constraint

Constraint text in OCL

Dresden OCL2 Toolkit (OCL2.0 based)

- Redesign of the Dresden OCL Toolkit based on the proposed metamodel for OCL 2.0
- Already available
  - **MOF repository** implementation
  - **Parser**
  - **Code generator** to generate Java expressions for the evaluation of WFRs (OCL invariants on meta models)
- Under Development
  - Full Java code generation
  - Application program instrumentation
  - SQL code generation

OCL in Business Rules Solutions

OCL in Business Rules Solutions

- Use of OCL in the UML-related world is recently growing because of the availability OCL tools
- Two-step process
  - 1st step: Specification of OCL constraints for documentation
  - 2nd step: Evaluation of OCL constraints for checking business rules
- First examples of using OCL
  - MetaBoss projects (metaboss.sourceforge.net, Australia)
  - pleXX framework (www.exxcellent.de/plexx, Germany)
  - Cemagref (www.cemagref.fr, Agricultural and environmental engineering, France)
  - Further own case studies with Dresden OCL tools

OCL Constraints vs. Business Rules (1)

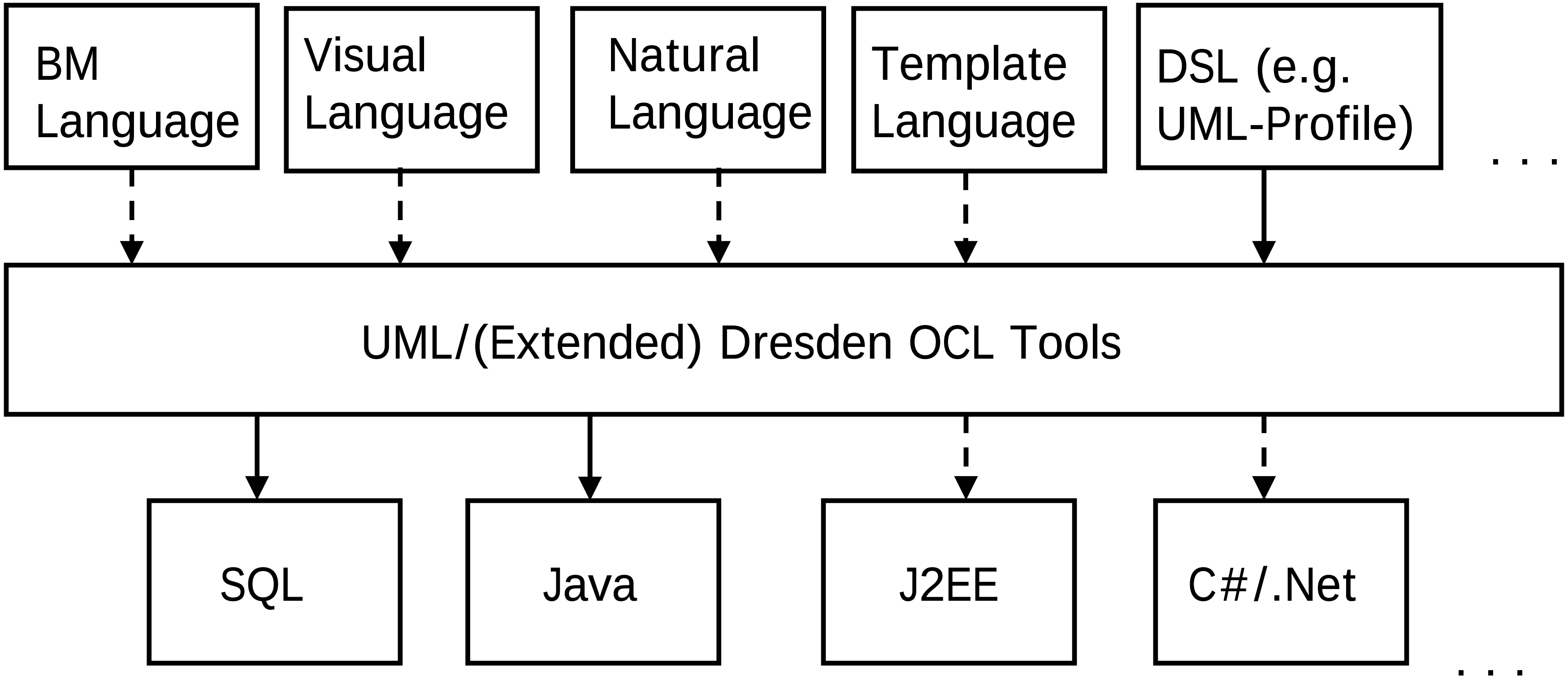
- Specification of Business Rules (BR)
  - At the **external** level: How can BR be expressed by the user in a simple and comprehensible way?
  - At the **conceptual** level: How can BR be represented inside the system?
  - At the **internal** level: How can BR be implemeted?
- Unanimous position in literature and reports
  - OCL is useful for the specification of BR
  - OCL is too difficult for business modelers
  - There are already different running OCL implementations

→ OCL is a good candidate for the conceptual level ☺

OCL Constraints vs. Business Rules (2)

- Consequences
  - Transformation of BR to OCL constraints
  - Evaluation OCL constraints by different implementation approaches
- Different approaches for user-friendly specification of BR, e.g.
  - Visual language for OCL (Constraint Trees, Kent et al 2002)
  - Business Modeling (BM) syntax for OCL (Octopus, Warmer et al 2003)
- Approaches for OCL implementations
  - DBMS based (SQL driven)
  - Application language driven (e.g. Java)

Dresden OCL Tools in a Business Rule Framework



Basic Types of BR and Their Mapping to OCL

Type of Business Rules	OCL concepts
Constraint	Invariant
Derivation rule	Derivation rule (derive) Query operation (body) (limited)
Reaction rule	Extended OCL (Actions, XOCL)

Scenario 1: Use a BR Language

Business Modeling (BM) Syntax for OCL as an example (Octopus, Warmer et al 2003)

- SQL like
- Supports all concepts of OCL
- User-friendly notation for predefined operations on collections and for predefined iterators
- Easy to implement in the framework of the Dresden OCL2 Toolkit because of the separation of concrete and abstract OCL syntax
- Further other concrete syntax possible

BM Example

One of our BR examples ...

- In standard OCL

```
context  Region::getNamesOfHotels:Bag(name)
body: hotelsInRegion->collect(name)
```

- In BM syntax of OCL

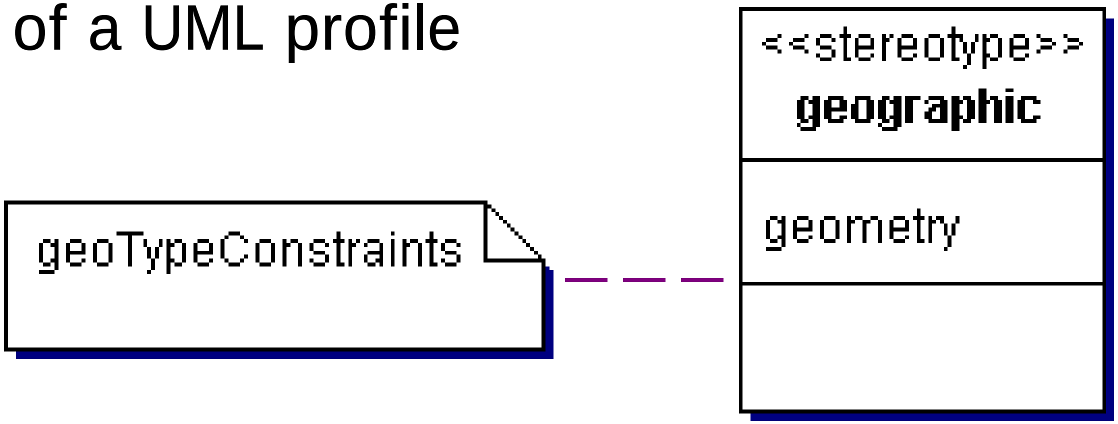
```
context Region::getNamesOfHotels:Bag(name)
body: collect h.name using h: Hotels
      from hotelsInRegion
```

Scenario 2:
Design a DSL

- Design of a Domain Specific Language (DSL) could be an approach to create a specialized BR language
- Generally valid built-in business rules
- Often used technique is the creation of a UML Profile
  - Stereotypes
  - Tagged values
  - (OCL) Constraints

Example: DSL for a Geographic IS (1)

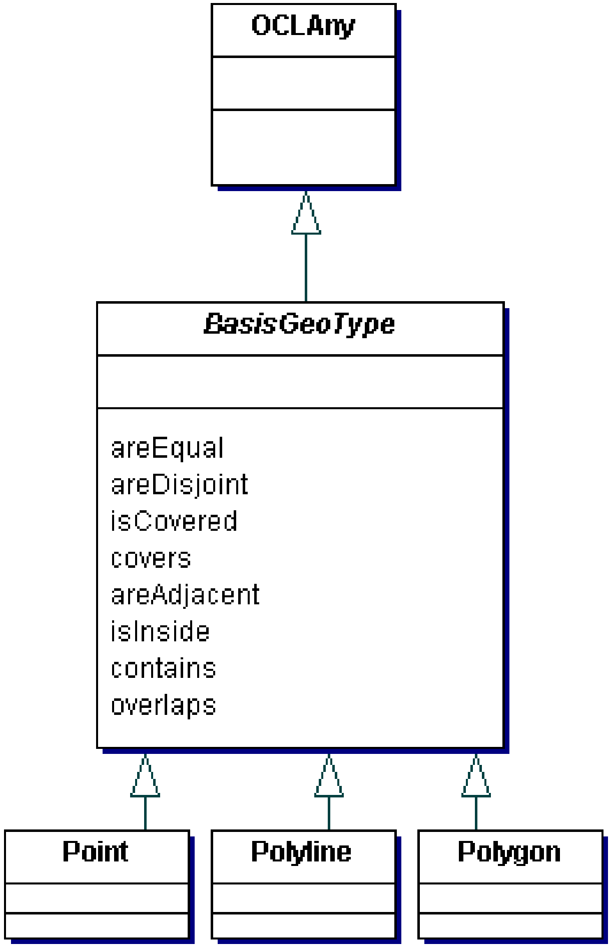
- DSL for GIS applications (Pinet et al, 2004)
- Definition of a UML profile



- Typical business rules of a GIS are „hidden“ in the profile
- Case study in an agricultural information system

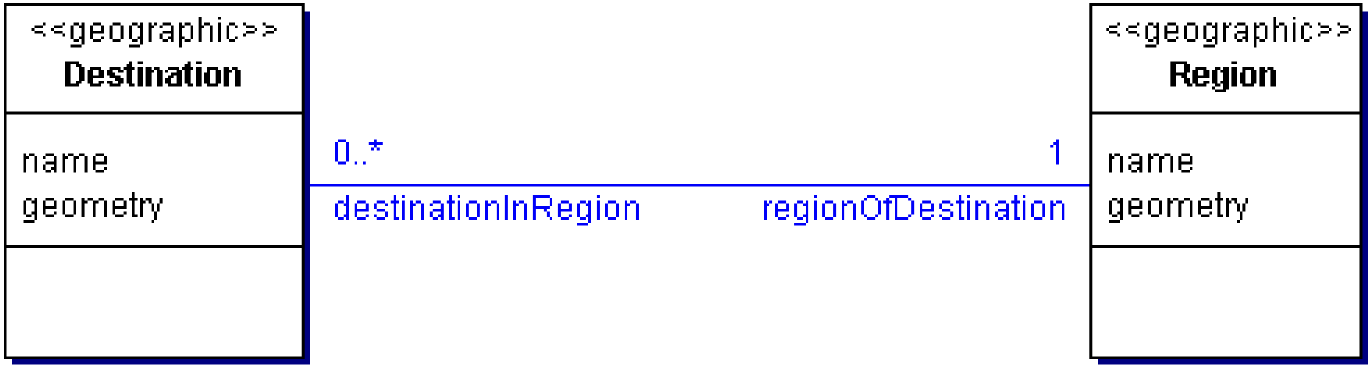
Example: DSL for a Geographic IS (2)

- „Spatial OCL“ for user-defined topological constraints
- Implemented by OCL2SQL



Example: DSL
for a Geographic IS (3)

- Embedding of our Hotel Reservation System into a GIS



- User-defined topological constraint

```
context Region
inv: destinationsInRegion ->
     forAll(self.geometry.contains(geometry))
```

TECHNISCHE UNIVERSITÄT DRESDEN

ST Softwaretechnologie

Lessons Learned and Conclusions

Lessons Learned (1)

- Our case studies show that the road to a broad use of OCL in the professional software development is still long.
- OCL is too difficult for most business modelers!
- BR/constraints should be executed, not only be documented!
- Subset of Business Rules can be implemented by OCL!
- Ideas are great, the realization of ideas by tools are better!

Lessons Learned (2)

Strengths of Dresden OCL Toolkit:

- Open source under the LGPL license
- Clean and small interfaces for extension of the toolkit and exchange of modules
- Code generation for Java and SQL
- Conformance to OCL1.3 (in future to OCL2.0)
- OCL Syntax assistant
- Metamodel based OCL20 architecture
- Allows implementations of OCL language extensions

Lessons Learned (3)

- Reuse in many research&development projects and in first business environments
- Maintenance at the university with minor ressources is difficult ☹
- We would be glad about further developers in the open source community ☺
- Feedback using our toolkit is welcome!

Conclusions

- There are two scenarios using OCL in business rules solutions:
  (1) Use a BR language and map the business rules internally to OCL
  (2) Design a DSL with internally hidden OCL constraints
- Extending OCL by actions covers a bigger set of business rules!
- Business rules projects should externally use OCL as long as no BR language exists!
- Dresden OCL Toolkit is reuseable for building tools which handles constraints in different forms

Outlook

- Maintenance of Dresden OCL13 Toolkit
- Further development of Dresden OCL20 Toolkit
- Further case studies
- Research questions:
  - How to extend OCL for higher expressiveness?
  - How to detect inconsistent specifications of constraints?
  - Can the Semantic Web (OWL and similar languages) benefit from OCL?
  - Clarification of the role of OCL in MDA and especially in the model-driven development of business rules applications