Consistency in UML

Presented by *Victoria Yang*

Overview

- Consistency Problems
- Frameworks of Consistency Checking
    - Constraint Checker
    - Pattern Checking

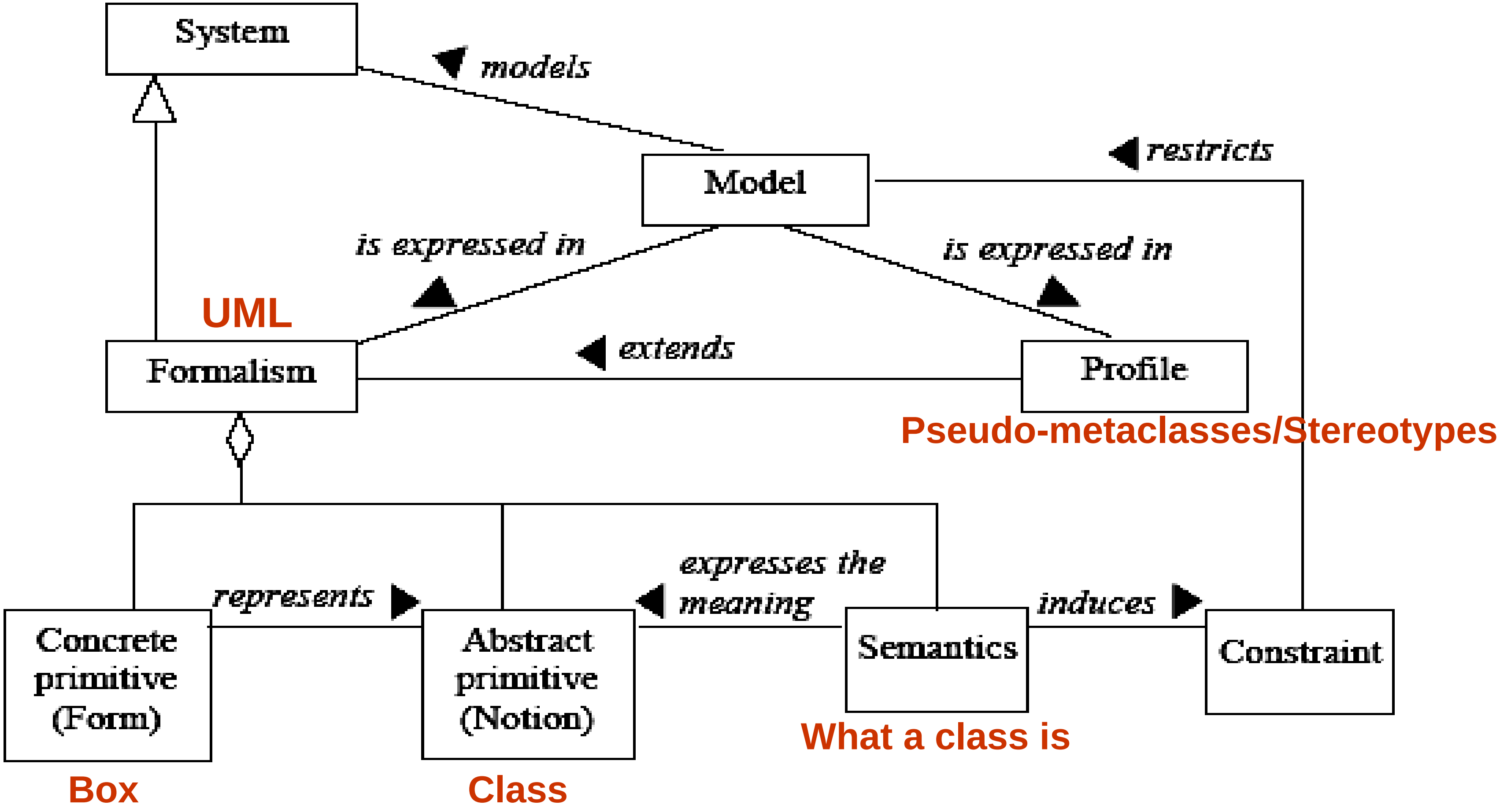
Categories of Consistency

- *Semantic* consistency
  - *Intra*-Consistency & *Inter*-Consistency

  OR

  - *Syntactic* & *Structural* consistency
- *Behavioral* consistency

Basic Notions in UML Model

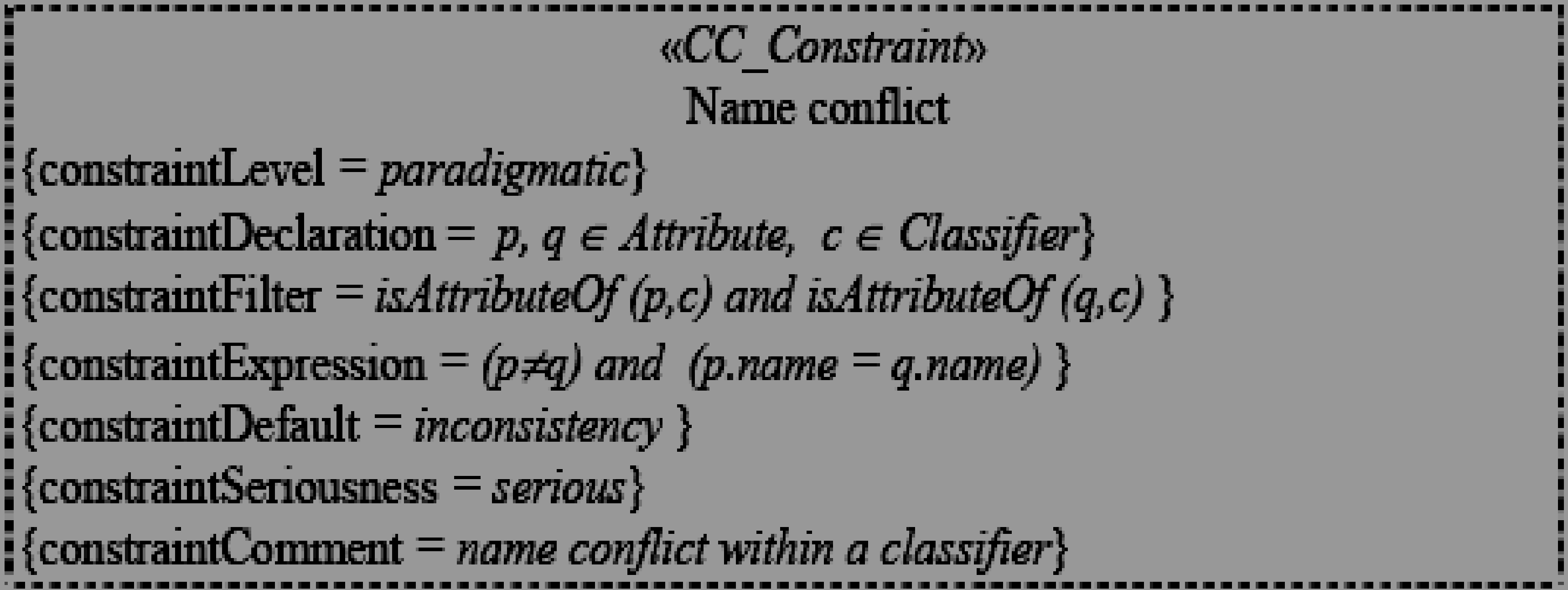


Framework I:
Constraint Checker (CC)

- Use Constraint Checker instead of OCL
- Map constraint to checking rule
  - *if* **Expression** *then* **Action**
- Extend checking rule with stereotypes

Example

- Well-formedness rule:
  - *No attributes may have the same name*



«CC_Constraint»

Name conflict

{constraintLevel = *paradigmatic*}

{constraintDeclaration = *p, q ∈ Attribute, c ∈ Classifier*}

{constraintFilter = *isAttributeOf (p,c) and isAttributeOf (q,c)* }

{constraintExpression = *(p≠q) and (p.name = q.name)* }

{constraintDefault = *inconsistency* }

{constraintSeriousness = *serious*}

{constraintComment = *name conflict within a classifier*}

Constraint Classification

1. *Paradigmatic* Level
2. *Paradigmatic Extension* Level
3. *Modeling Process* Level
4. *Target* or *Modeled Domain* Level
5. *Platform Specific* or *Implementation* Level

Paradigmatic Level

- Constraints from formalism.
  - Unchangeable semantic constraints.
  - Ideally checked automatically.
- Includes:
  - Class diagrams
  - Well-formedness rules
  - Semantics in natural language

Paradigmatic Level Cont.

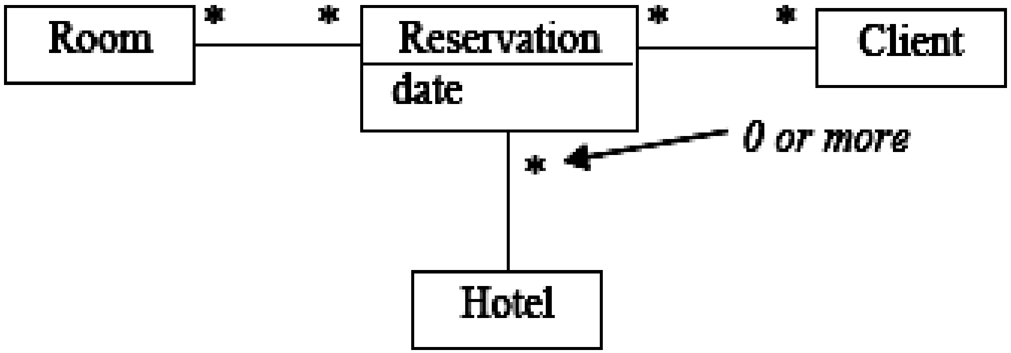
- Problems
  - Tool insufficiency
    - Tools not enforce some rules.
  - Unexpressed Rules
    - Semantics not in UML
  - Semantic Rules
    - In natural language & not expressible in OCL

Paradigmatic Extension Level

- *Changeable* constraints from *stereotypes*
- Problems:
    - Stereotypes
        - Unspecified in UML
    - Tagged Value
        - User specified

Modeling Process Level

- Style Guide
  - Multiple representation of one system.
  - Rules to choose best representations.
  - Implicit constraints



Target Domain Level

- Objects from Real World
  - Protocol dependant connection rules.
- Constraints are dynamic
  - Only checked at runtime.

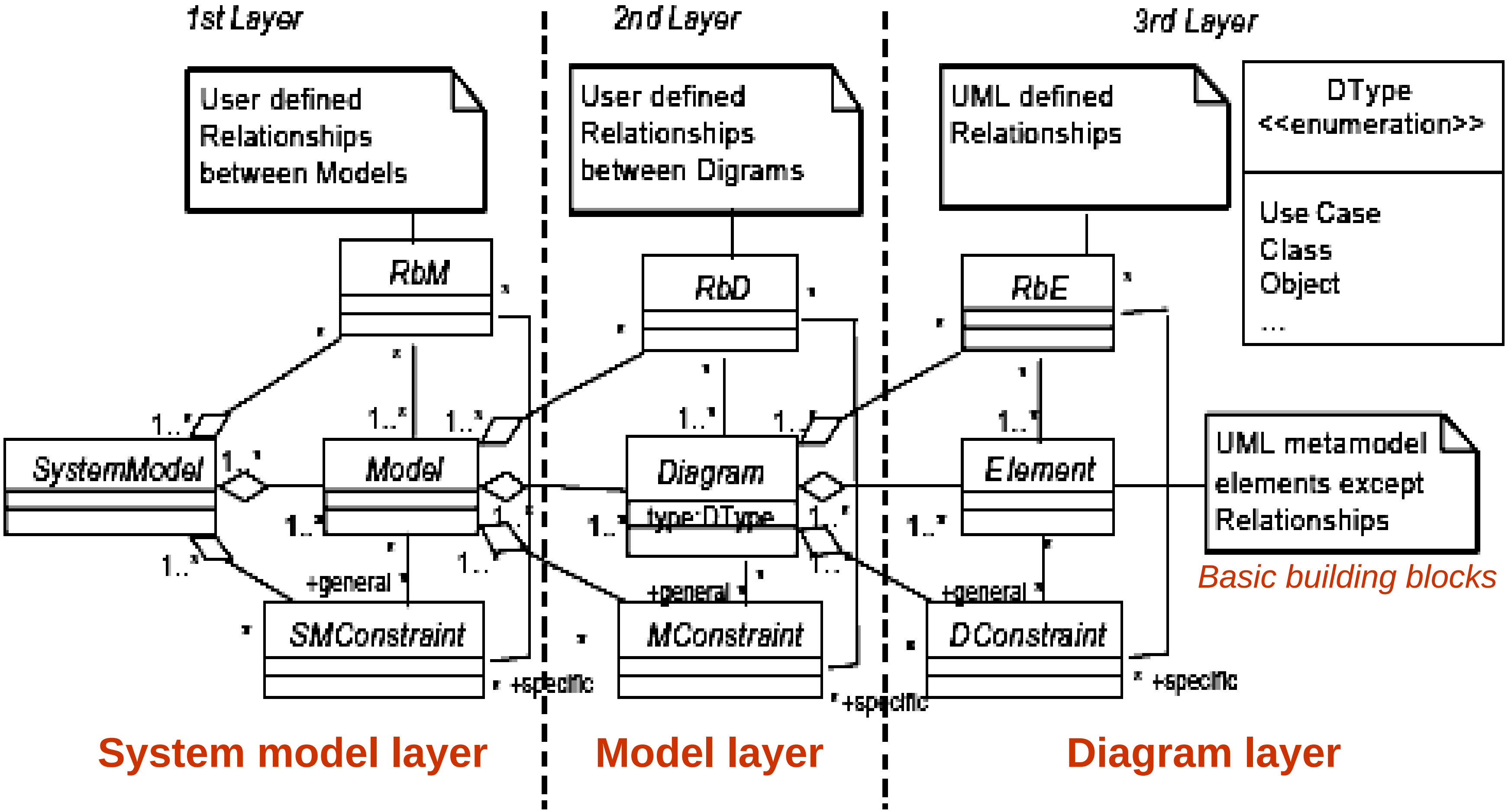
Implementation Level

- Includes:
  - Target Language Constraints
    - Java: multiple inheritance forbidden.
  - Implementation Constraints
    - From language or implicit implementation rules
    - Build car first or wheels first.
  - Description Policy
    - Policy: All constraints described in UML

Framework II: Pattern Checking

1. Specification phase
2. Algorithmization phase
3. Checking phase

System Model Structure



System model layer

Model layer

Diagram layer

Basic building blocks

Specification Phase

- *Define* consistency for specific development process
- Customize for Development Process
  - Diagrams and relations between them.
  - Constraints on the diagram and relations between diagrams.
- Result: Specialized System Metamodel

Algorithmization Phase

- *Formulate* set of consistency checking procedures.

- Virtual Diagram
  - Composition of diagrams of same type
  - Check intra-consistency

Checking Phase

- *Apply* consistency checking procedures to concrete software project.

Constraint Checker (CC) vs. Pattern Checking

- **Specification** Phase
  - 5 semantic levels
- **Algorithmization** Phase
  - Extract rules for each level.
- **Checking** Phase
  - Apply CC on rules.

References

- Jean Louis Sourrouille, Guy Caplat: *Checking UML Model Consistency,* Workshop on Consistency Problems in UML-based Software Development, 2002

- Bogumila Hnatkowska, Zbigniew Huzar, Ludwik Kuzniarz, Lech Tuzinkiewicz: *A systematic approach to consistency within UML based software development process*, Workshop on Consistency Problems in UML-based Software Development, 2002

- John Derrick, David Akehurst, and Eerke Boiten: *A framework for UML consistency*, *<<UML>> 2002 Workshop on Consistency Problems in UML-based Software Development*, pages 30-45, October 2002.

Questions