A UML Class Diagram Analyzer

Tiago Massoni

Rohit Gheyi

Paulo Borba

Software Productivity Group

Informatics Center – UFPE

October 2004

UML and Critical Systems

- UML
  - Growing interest
  - Explore concepts
  - Address important problems
  - OCL to specify complex constraints
- Complex structures with class diagrams

UML/OCL and Tools

- Lack of semantics
  - Some have proposed approaches
  - Did not stimulate tools for automatic analysis

- Absence of tool support
  - Additional trouble in critical systems
  - Structural modeling errors are hard to detect

Subtle Errors when Modeling Critical Systems

- Structural errors
  - OCL invariants may turn a class diagram over-constrained or inconsistent
  - Under-constrained diagrams allow incorrect implementations

- These problems are desirable to be automatically detected…

Contributions

- Approach for automatic analysis of UML class diagrams
  - A precise semantics for class diagrams is given

- Semantic model: Alloy
  - Object modeling language
  - Analysis tool for concrete feedback
  - Modeled several critical systems

Contributions

- Semantics by mapping
  - Mapping rules from diagrammatic and OCL class invariants to Alloy

- We leverage automatic analysis of Alloy to class diagrams
  - Automatic generation of snapshots
  - Assertion checking

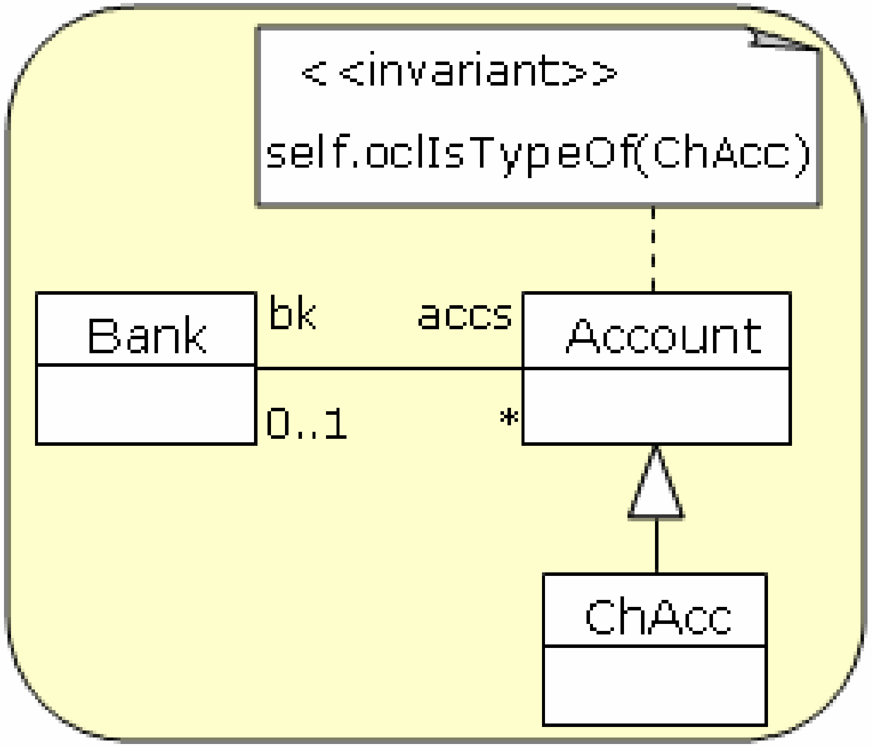
Outline

- Alloy
- Semantics for UML class diagrams
- Example
- Alloy in Critical Systems

Alloy

- MIT – Software Design Group (Daniel Jackson)
- Simple language for declarative modeling
  - Primarily structural properties
  - Sets, relations and predicate logic
- Alloy Analyzer

UML to Alloy



```
fact BankProperties {
  Account = ChAcc
  all a:Account|lone a.~accs
  bk = ~accs
}
sig Bank {
  accs: set Account
}
sig Account{
  bk: set Bank
}
sig ChAcc extends Account {}
```

Semantics for Class Diagrams

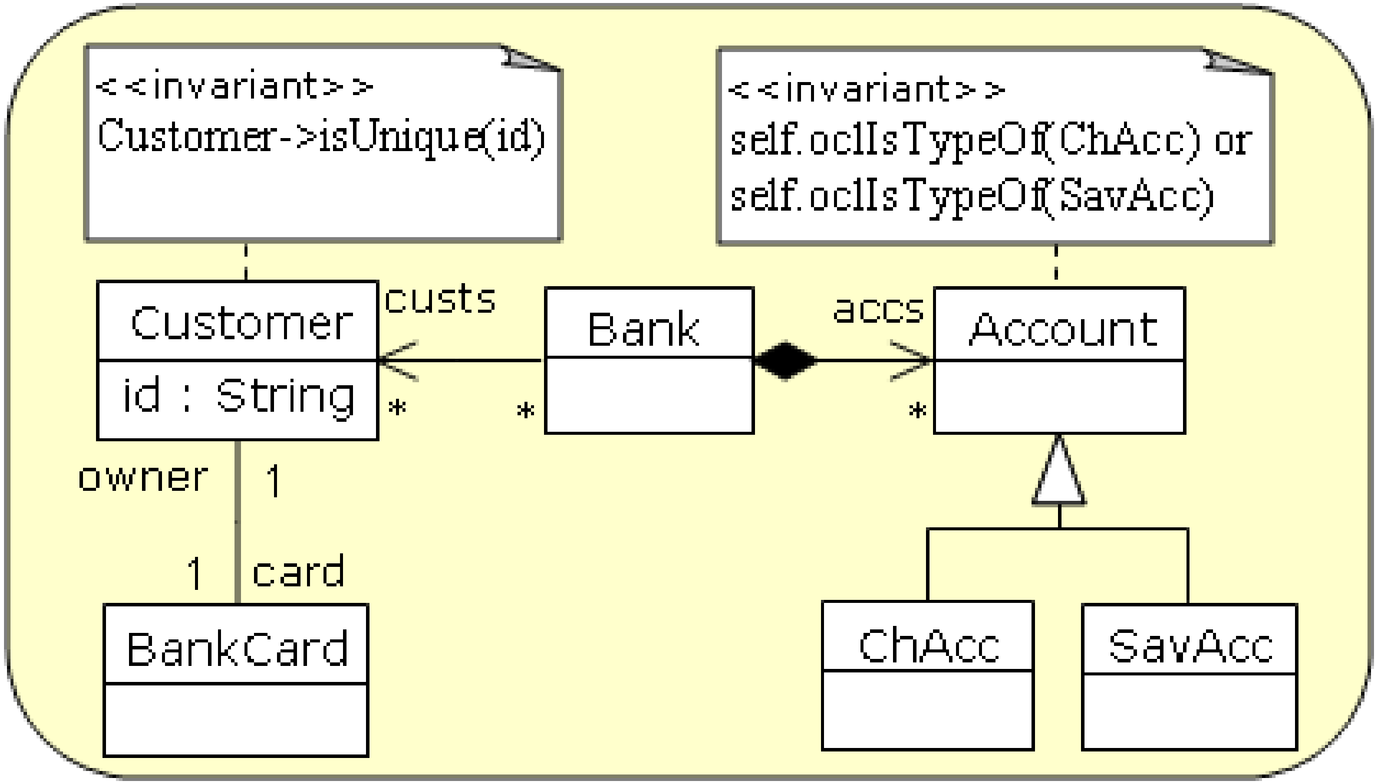
- Initial focus on structural properties
  - Avoided constructs with undefined semantics
- Diagrammatic constructs
  - Classes and interfaces: signatures
  - Binary associations and attributes: relations
  - Generalization: extends

Semantics for Class Diagrams

- OCL invariants: Alloy facts
  - Universally quantified on self

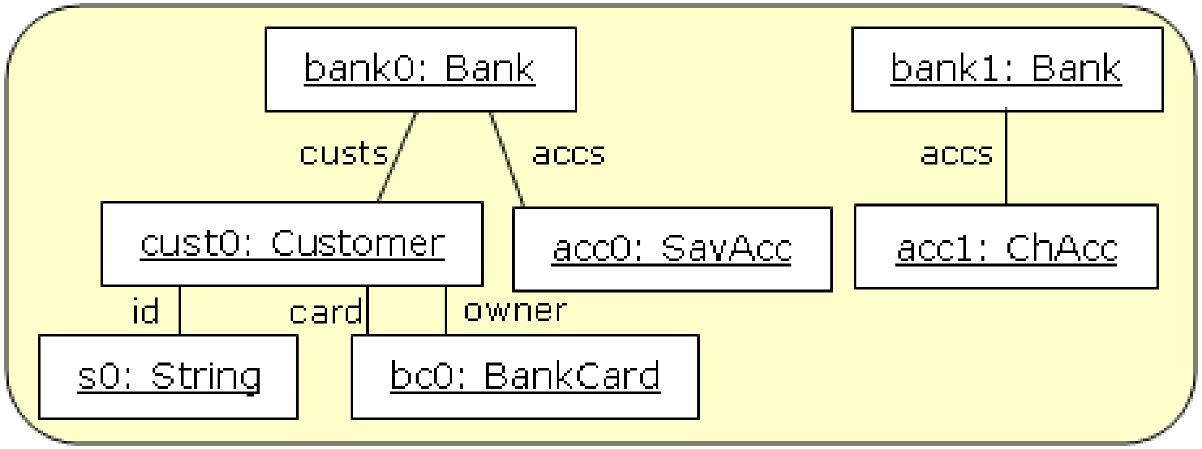
OCL	Alloy
X.oclIsTypeOf(Y)	X in Y
X.allInstances	X
X->isEmpty()	no X
X->forAll(a|P)	all a:X| P
X->size()	#X

Analysis Example



Analysis Example

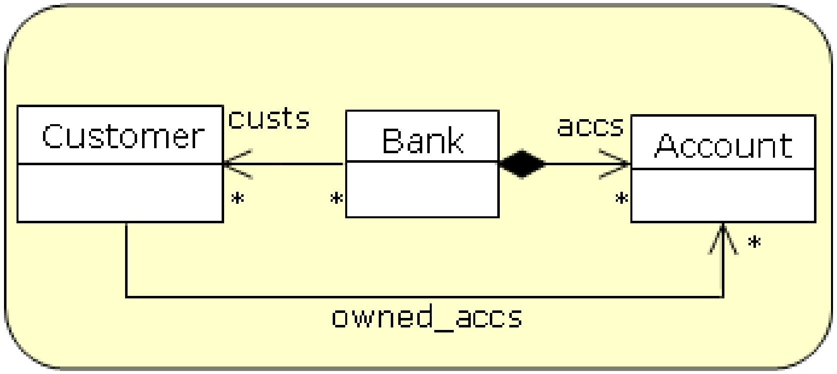
Snapshot 1 : scope of two

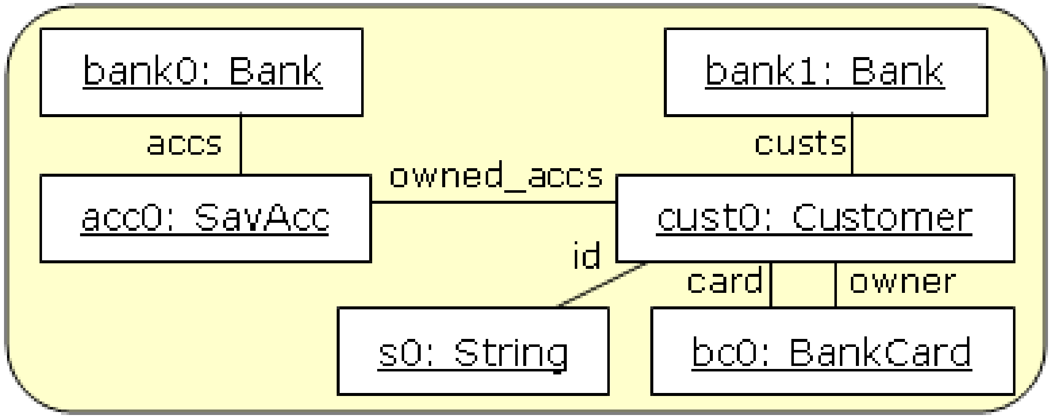


Customers and their personal accounts aren't related at all (they could be in different banks)

Analysis Example

Changing the diagram



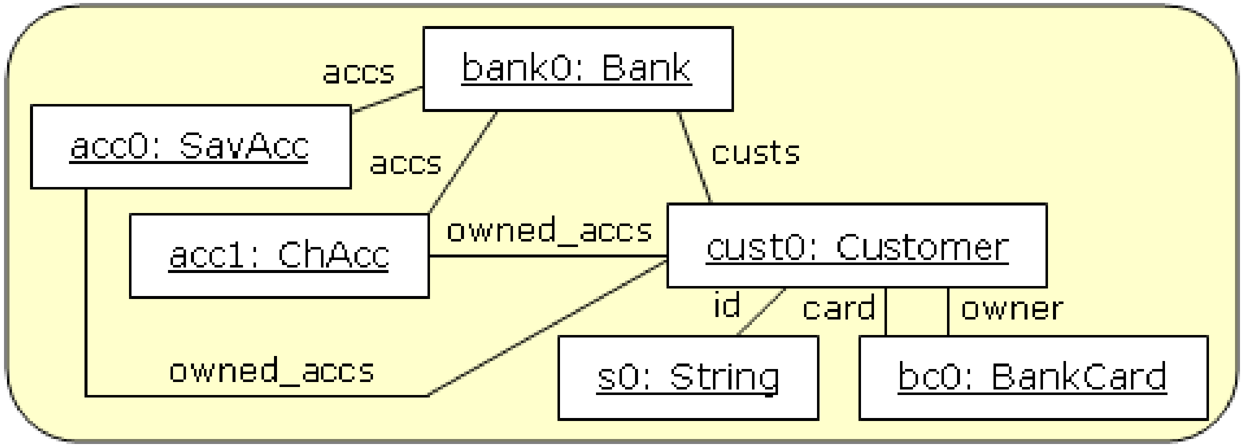


Snapshot 2 (still under-constrained)

Analysis Example

Adding an OCL constraint:

```
context Bank inv customersAccountsInBank:
self.custs.owned_accs->includes(self.accs)
```



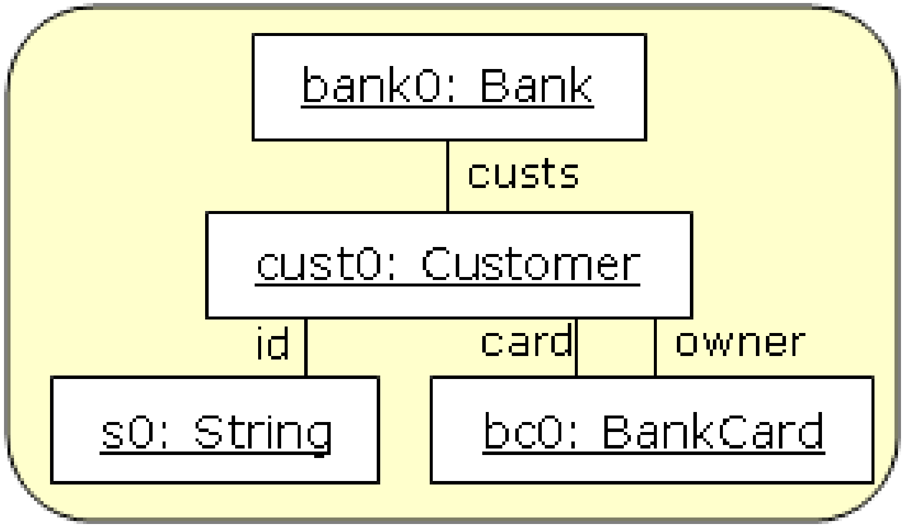
Snapshot 3
(great!)

Analysis Example

- I'd like to check whether every customer with cards has an account
  - Within the Customer context:

```
self.card->notEmpty() implies self.owned_accs->notEmpty()
```

- Counterexample:



Applications of Alloy in Critical Systems

- Radiation Therapy Machine
  - Operation Commutativity
- Railway System
- Access Control
- Air-traffic control

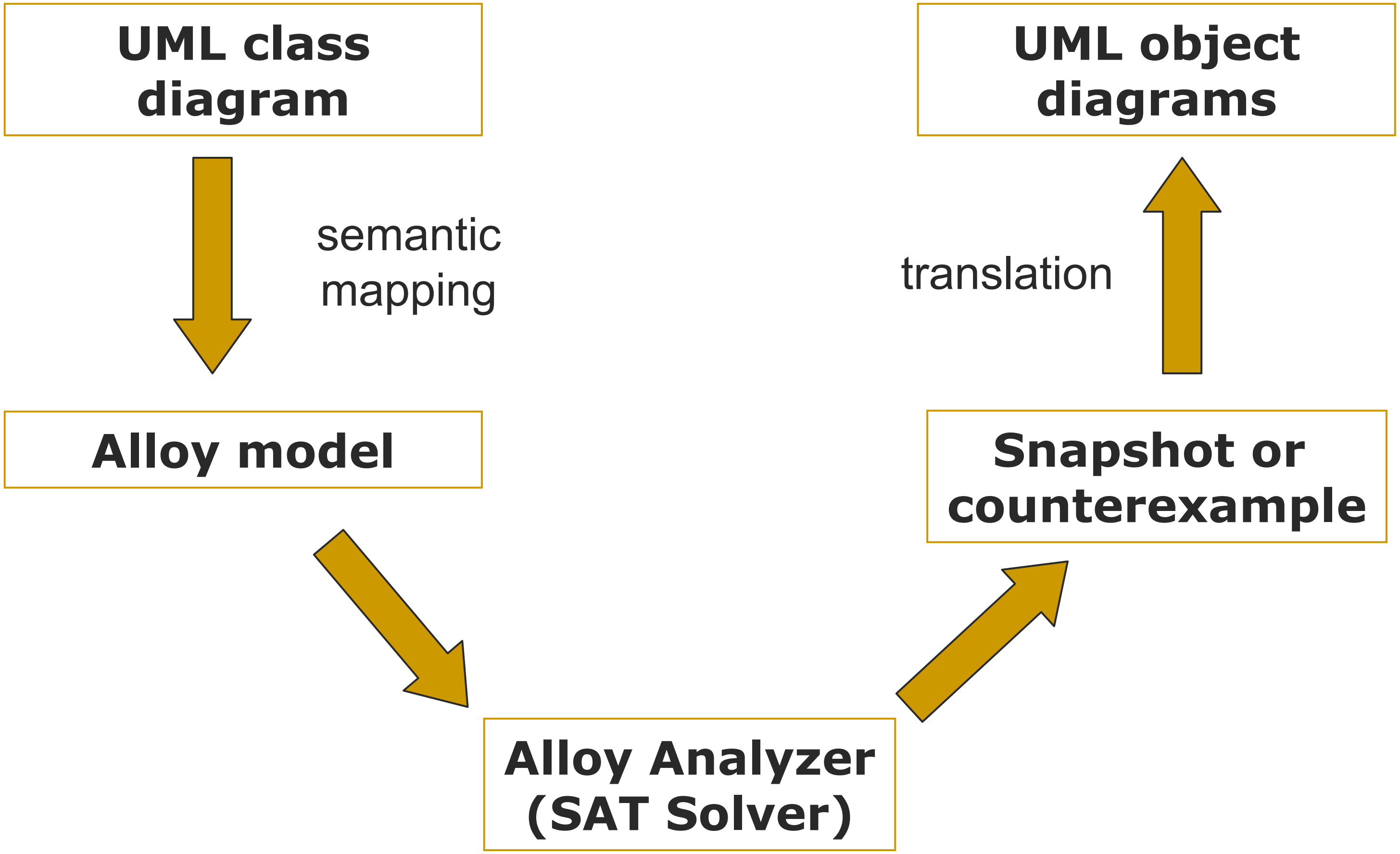
Conclusion

- Visual identification of modeling problems
- Covering many more states than any testing tool
- Leverage the benefits to UML Class Diagrams
- Future Work
  - Prototype (translation, analysis)
  - Behavioral Modeling
  - Case studies
  - Denotational semantics for class diagrams
  - Equivalence notion for models

Software Productivity Group

- www.cin.ufpe.br/spg
  - Model refactoring
  - Synchronization model-source code
  - Semantics
  - Formal Methods

Putting Analysis to Work



UML class diagram

semantic mapping

Alloy model

Alloy Analyzer (SAT Solver)

Snapshot or counterexample

translation

UML object diagrams

Alloy Analyzer

- Two kinds of analysis
  - Simulation
  - Assertion checking
- Analysis
  - Bounded by a scope of objects and relations

A UML Class Diagram Analyzer

Tiago Massoni

Rohit Gheyi

Paulo Borba

Software Productivity Group

Informatics Center – UFPE

October 2004