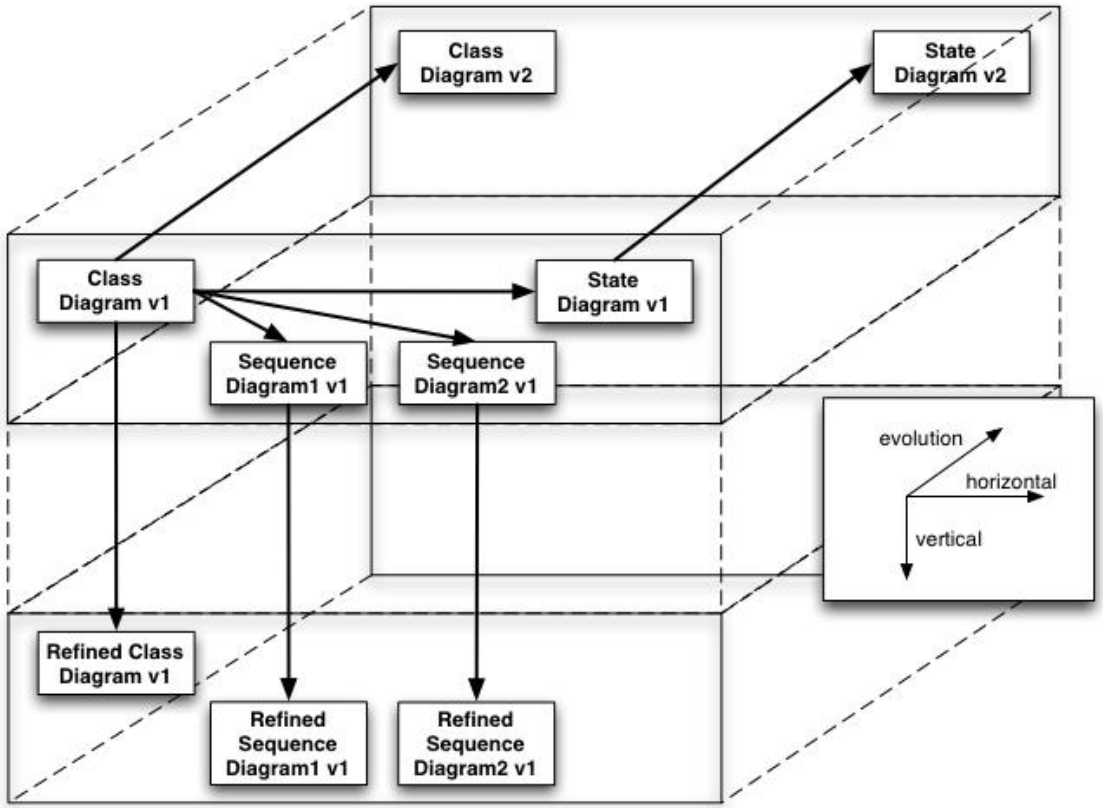
Consistency Maintenance

Ragnhild Van Der Straeten
SSEL, Vrije Universiteit Brussel
Brussel

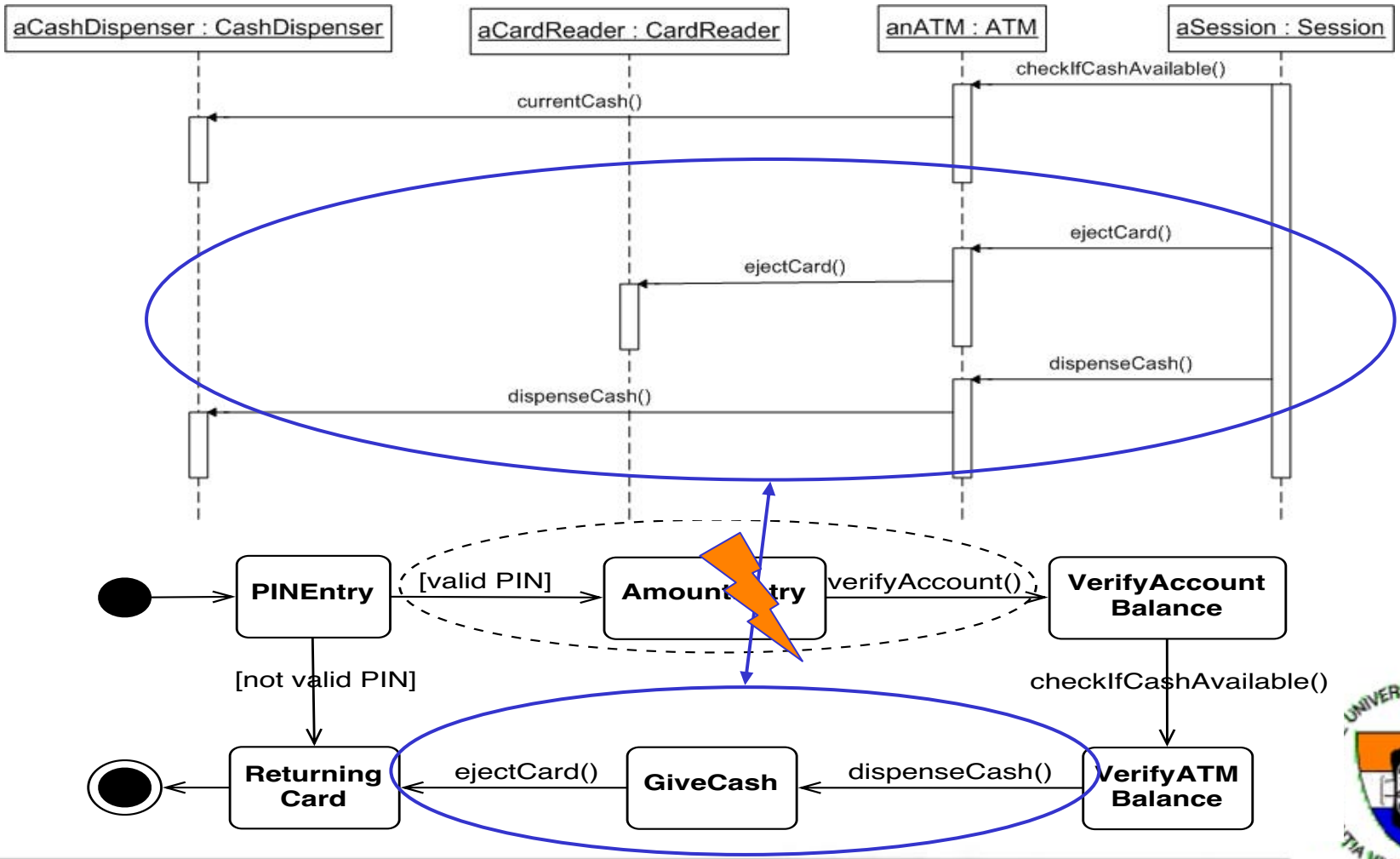
Kinds of Consistencies



Classification of model inconsistencies

	Behavioural	**Structural**
Specification – specification conflicts	•Invocable inconsistency •Observation inconsistency	•Dangling type reference •Inherited association inconsistency •Role specification missing
Specification - instance conflicts	•Incompatible definition	•Instance specification missing
Instance - instance conflicts	•Invocation inconsistency •Observation inconsistency •**Incompatible behaviour**	•Disconnected model

Example


aCashDispenser : CashDispenser
aCardReader : CardReader
anATM : ATM
aSession : Session
checkIfCashAvailable()
currentCash()
ejectCard()
ejectCard()
dispenseCash()
dispenseCash()
PINEntry
[valid PIN]
verifyAccount()
VerifyAccount Balance
[not valid PIN]
checkIfCashAvailable()
Returning Card
ejectCard()
GiveCash
dispenseCash()
VerifyATM Balance

Formalism for consistency management

- consistency management requires:
  - a decidable formalism (to detect inconsistencies)
    - detection of inconsistencies requires answering queries over sets of individuals
  - a generic framework for consistency detection/resolution
    - to facilitate adding/removing/modifying consistency rules

Why description logics?

- decidable two-variable fragment of first-order predicate logic
- consistency between metamodel and models is guaranteed for free
- straightforward mapping of UML metamodel
  - classes $\rightarrow$ concepts
  - associations $\rightarrow$ roles
  - attributes $\rightarrow$ roles or concrete domain attribute
  - inheritance $\rightarrow$ subsumption mechanism and transitive closure
- detecting inconsistencies = answering queries

But, there is more...

- Wide range of software artefacts:
  - requirements, architectures, design models, source code, tests
- Evolution
  - e.g. refactoring
- Refinement

??Other inconsistencies??

Other approaches...

- Graph transformation schemes (e.g. by Bottoni *et al.*)
- Model for change propagation based on graph rewriting (e.g. by Rajlich)
- Prolog, SOUL (e.g. by R. Wuyts and K. Mens)

??Requirements which must be met by an approach??

??Does THE approach exist??

??Tool support??