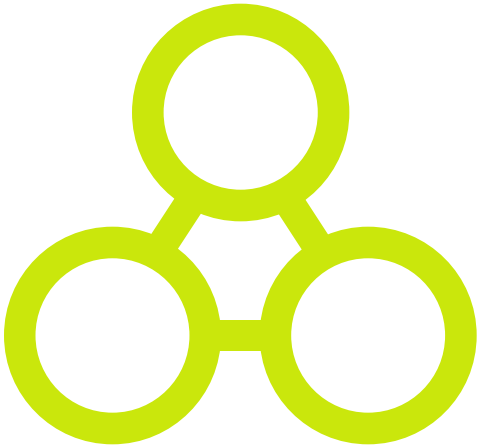
Transforming OntoUML into Alloy: Towards conceptual model validation using a lightweight formal method



nemo

ontology & conceptual modeling research group

Bernardo F.B. Braga, João Paulo A. Almeida, Alessander Botti Benevides and Giancarlo Guizzardi

http://nemo.inf.ufes.br

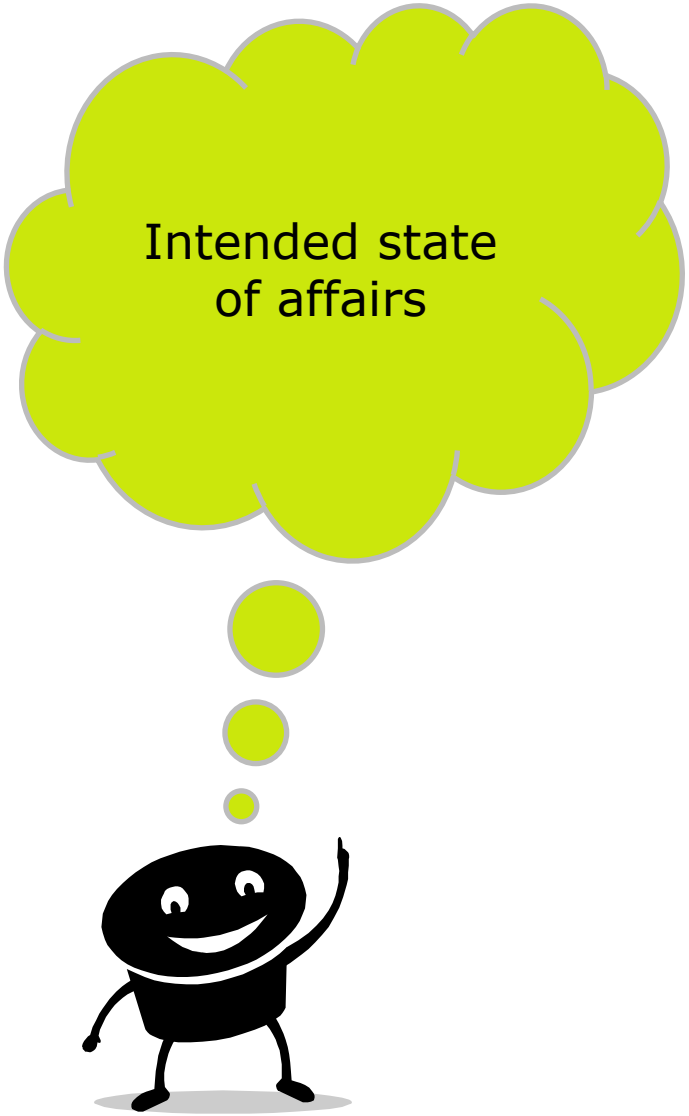
Computer Science Department

Federal University of Espírito Santo

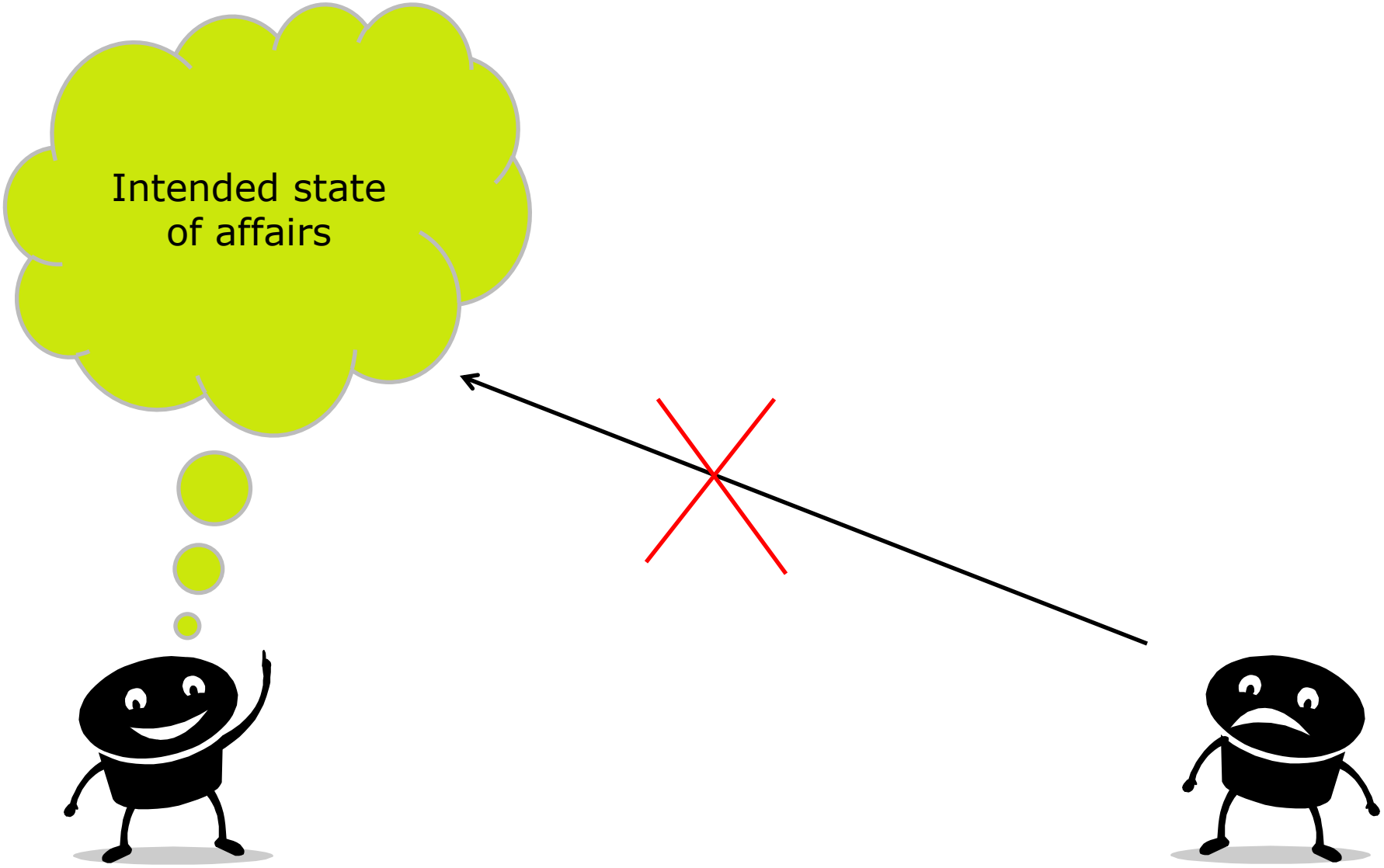
Motivation

- Conceptual models as a “point of view”
  - **Represent the modeller’s intended conceptualization**
- The model should ideally:
  - **describe all states of affairs that are deemed admissible and**
  - **rule out those deemed inadmissible**
  - (according to the conceptualization)
- Assessing the quality of conceptual models is key to ensure that conceptual models can be used effectively as a basis for understanding, agreement and construction of information systems.

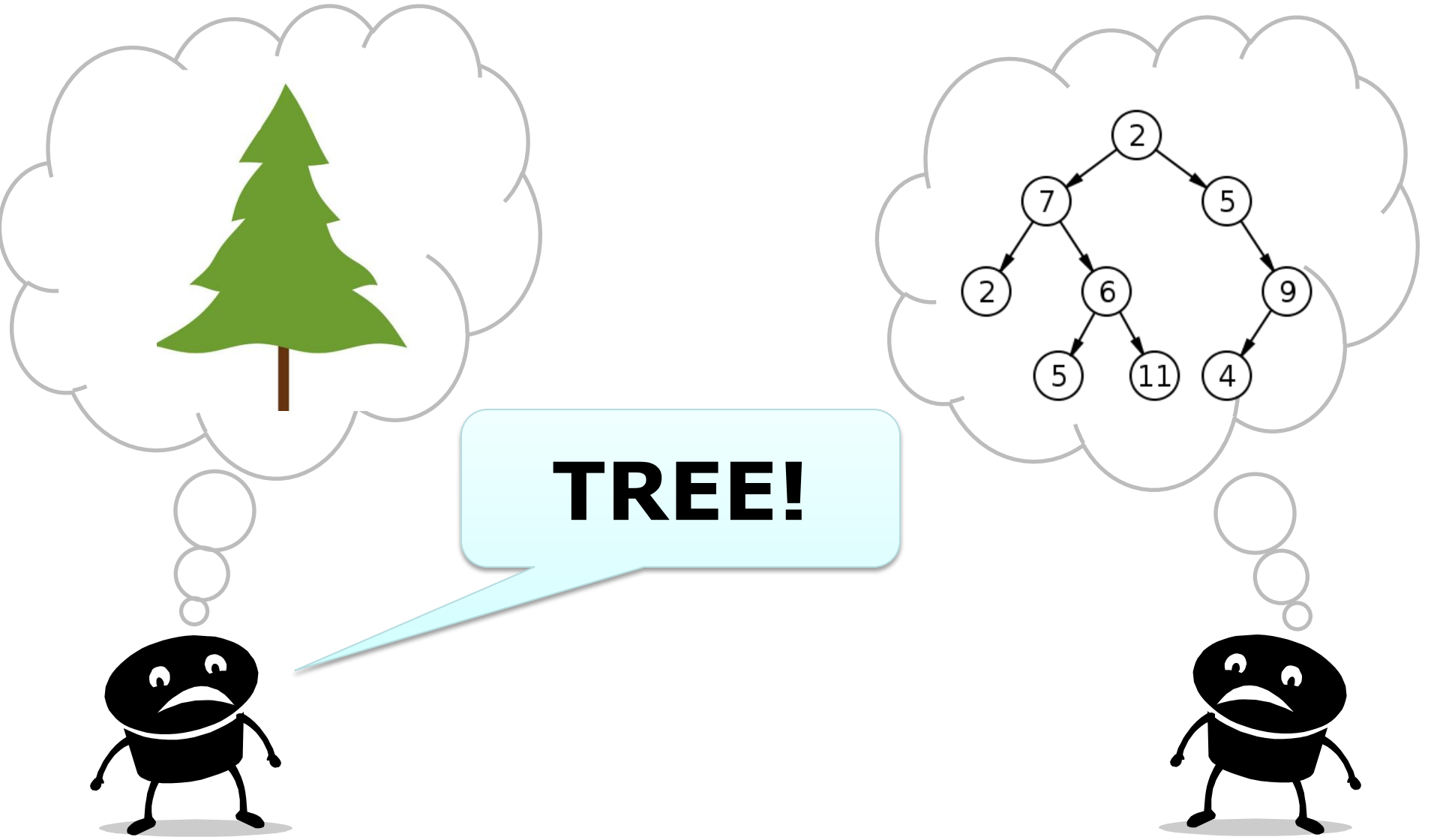
Conceptualization



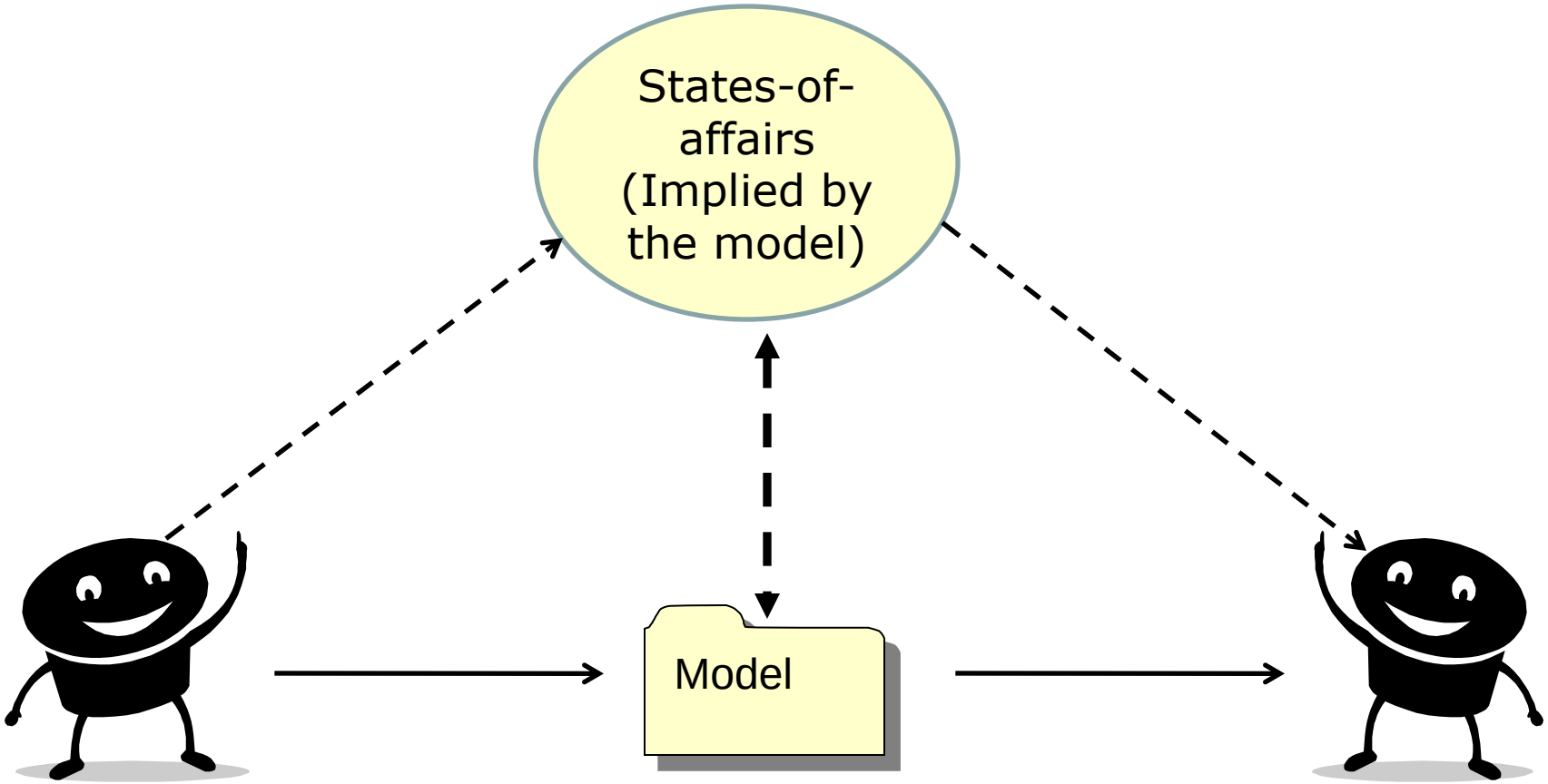
Communicating



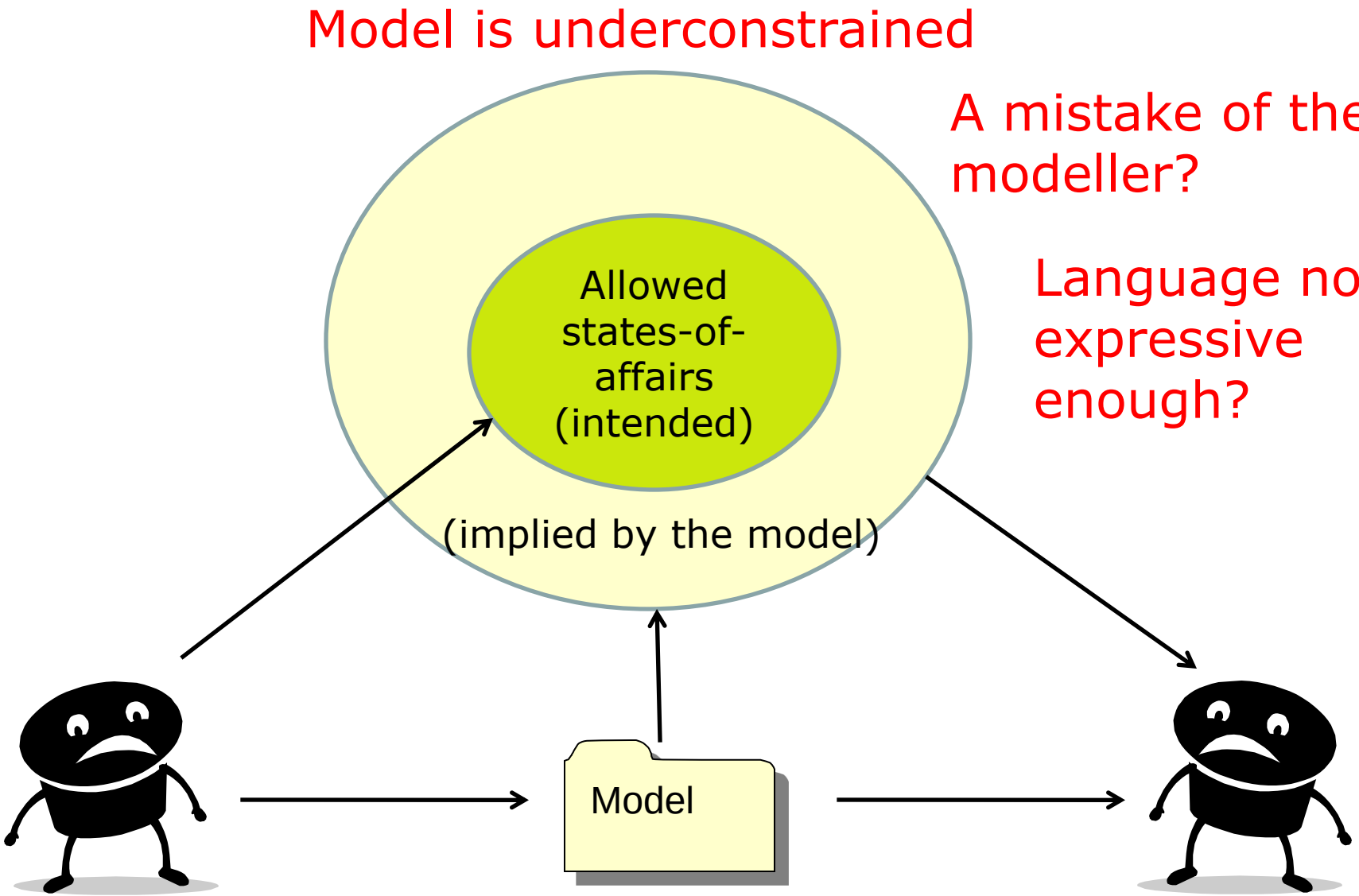
Communicating with natural language



Communicating with a model

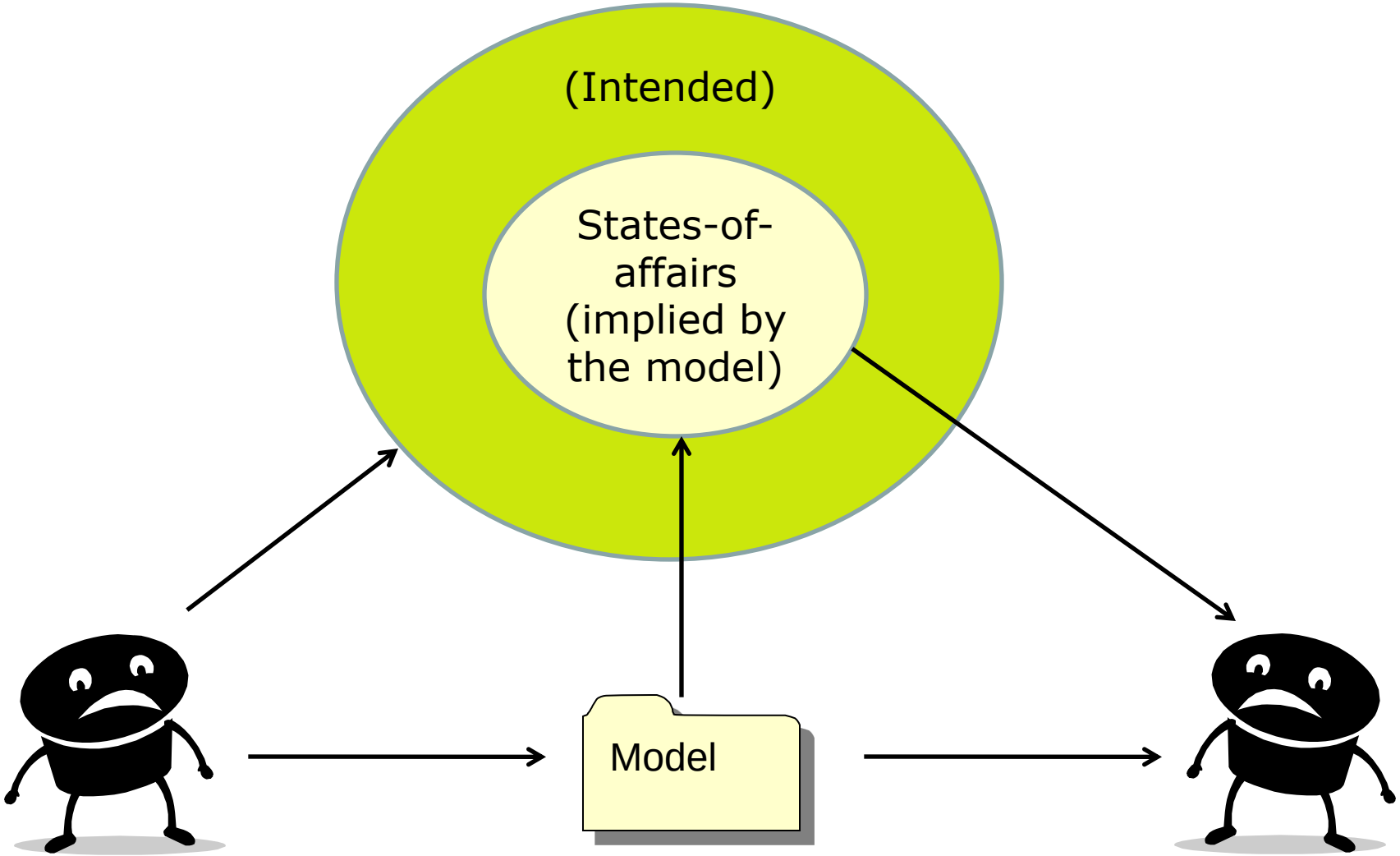


Intended x Modeled States-of-Affairs



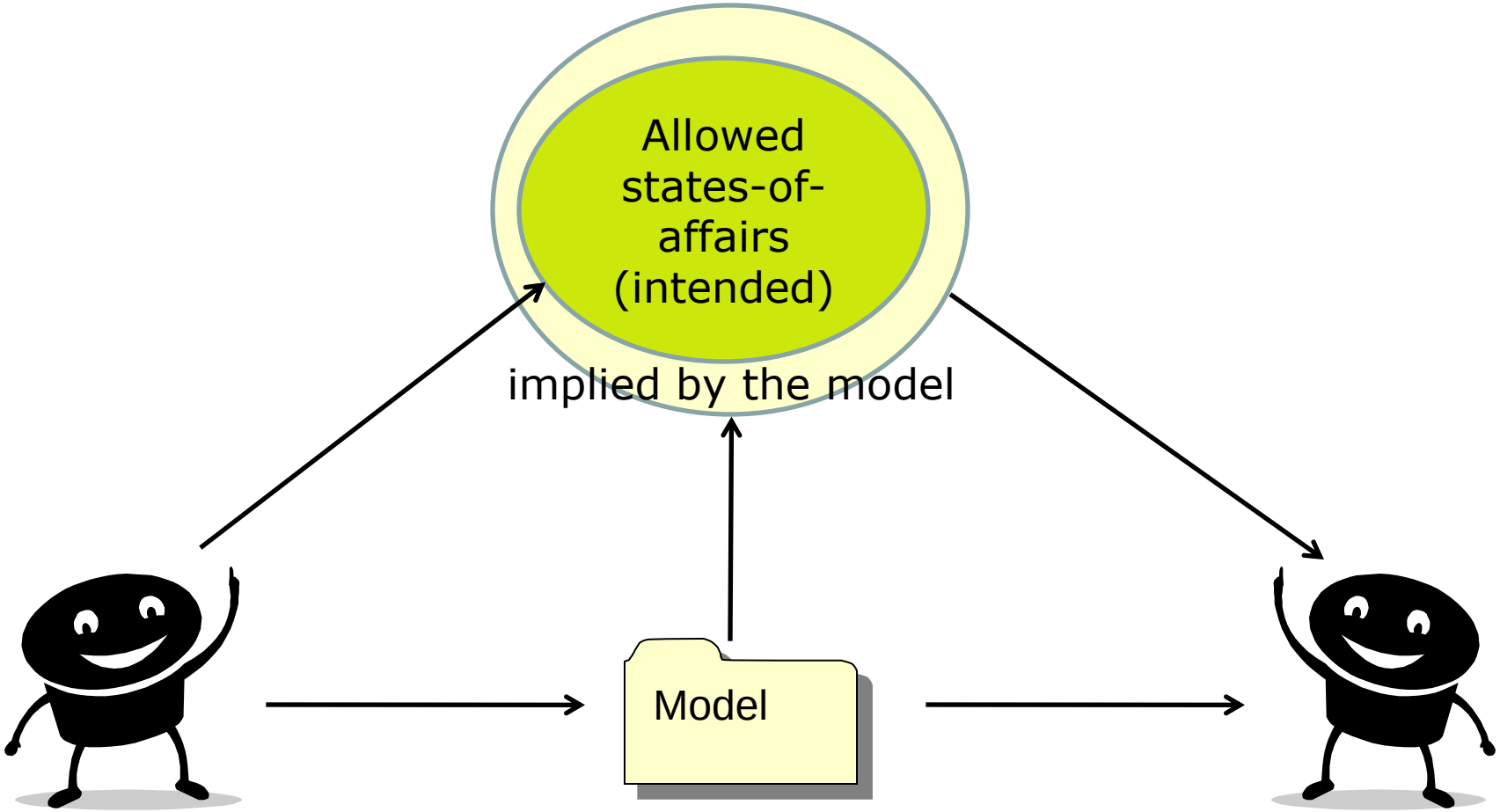
Intended x Modeled States-of-Affairs

Model is overconstrained

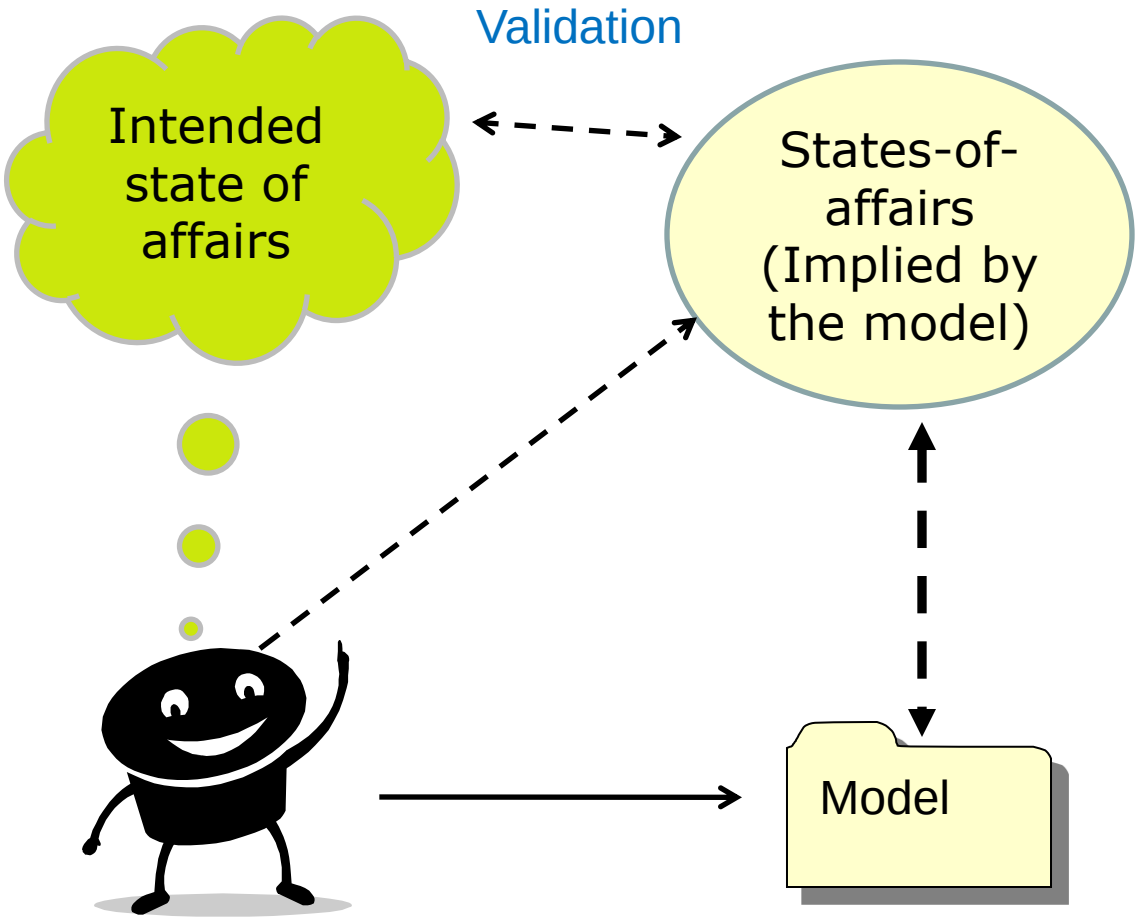


Intended x Modeled States-of-Affairs

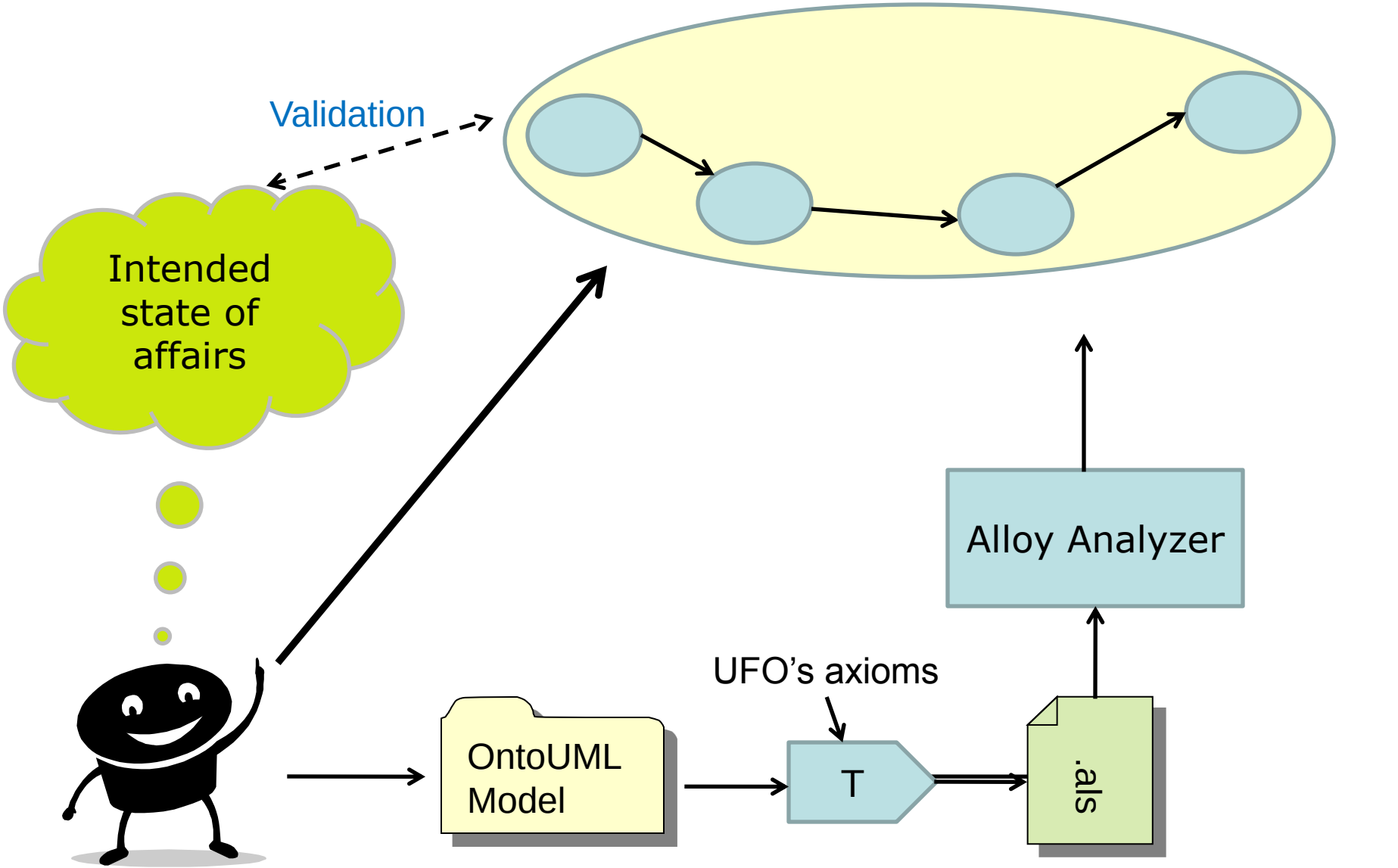
High precision
And coverage



Validating a model



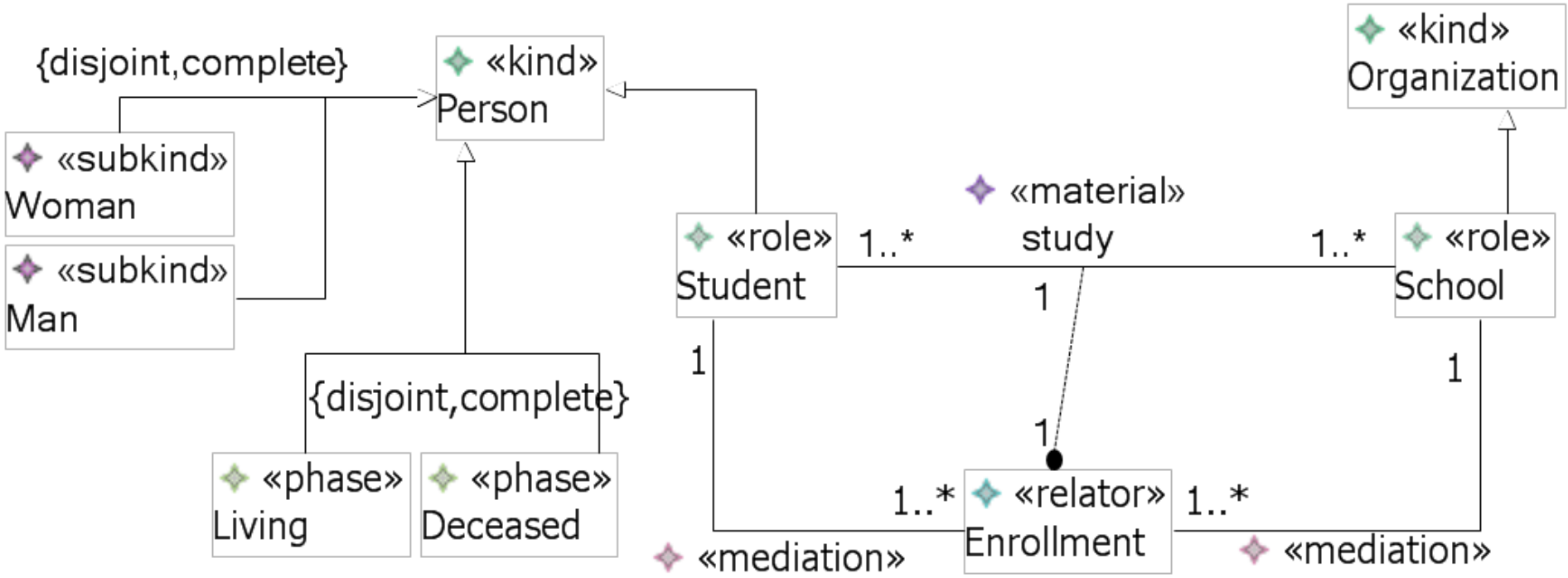
Our Approach: Transform OntoUML Model into Logic-Based Alloy for Analysis



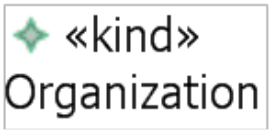
OntoUML

- UML profile that incorporates the theory in the UFO foundational ontology
  - **finer-grained distinctions between different types of classes (among other things)**
  - basis on philosophy, psychology, empirical evidence
- Modal meta-properties for object classifiers
  - Distinguishing rigid, semi-rigid and anti-rigid classifiers
    - (and therefore distinguish properties that apply necessarily to objects from those that apply contingently)

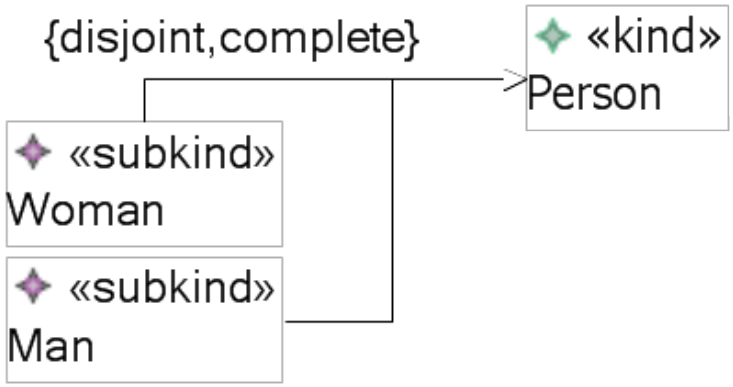
An Example


{disjoint,complete}
«kind»
Person
«kind»
Organization
«subkind»
Woman
«subkind»
Man
«material»
study
«role»
Student
1..*
1..*
«role»
School
1
1
1
1
{disjoint,complete}
«phase»
Living
«phase»
Deceased
1..*
«relator»
Enrollment
1..*
«mediation»
«mediation»

An Example

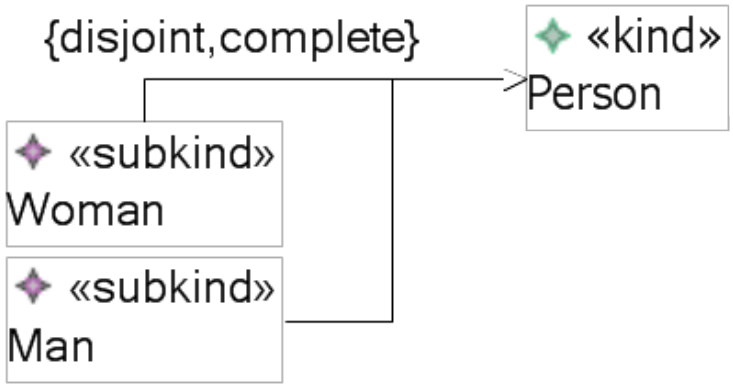


An Example



«kind»
Organization

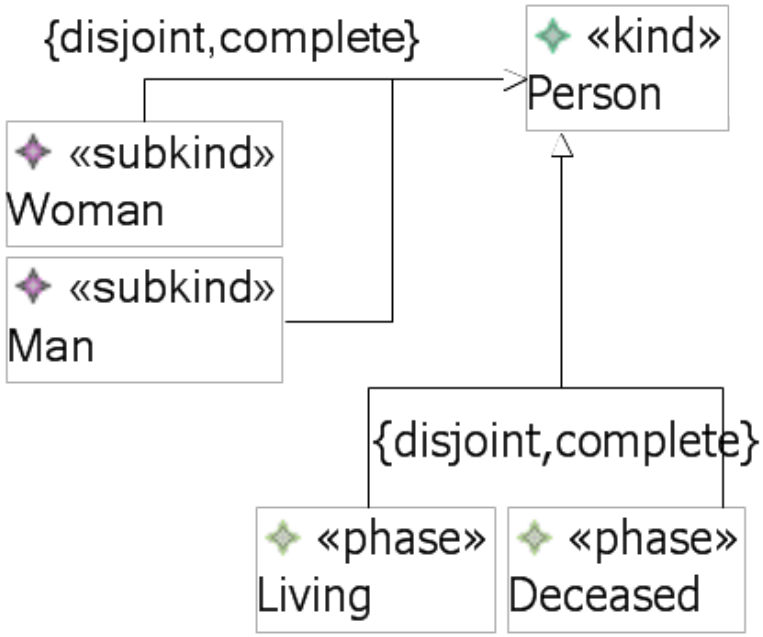
An Example



«kind»
Organization

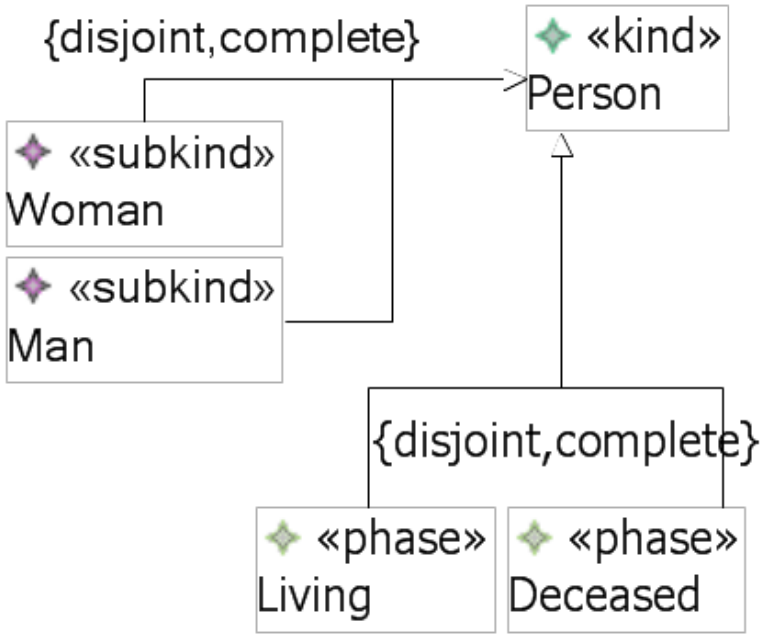
$$Rigid(T) \equiv \Box(\forall x(T(x) \rightarrow \Box(\varepsilon(x) \rightarrow T(x))))$$

An Example



«kind»
Organization

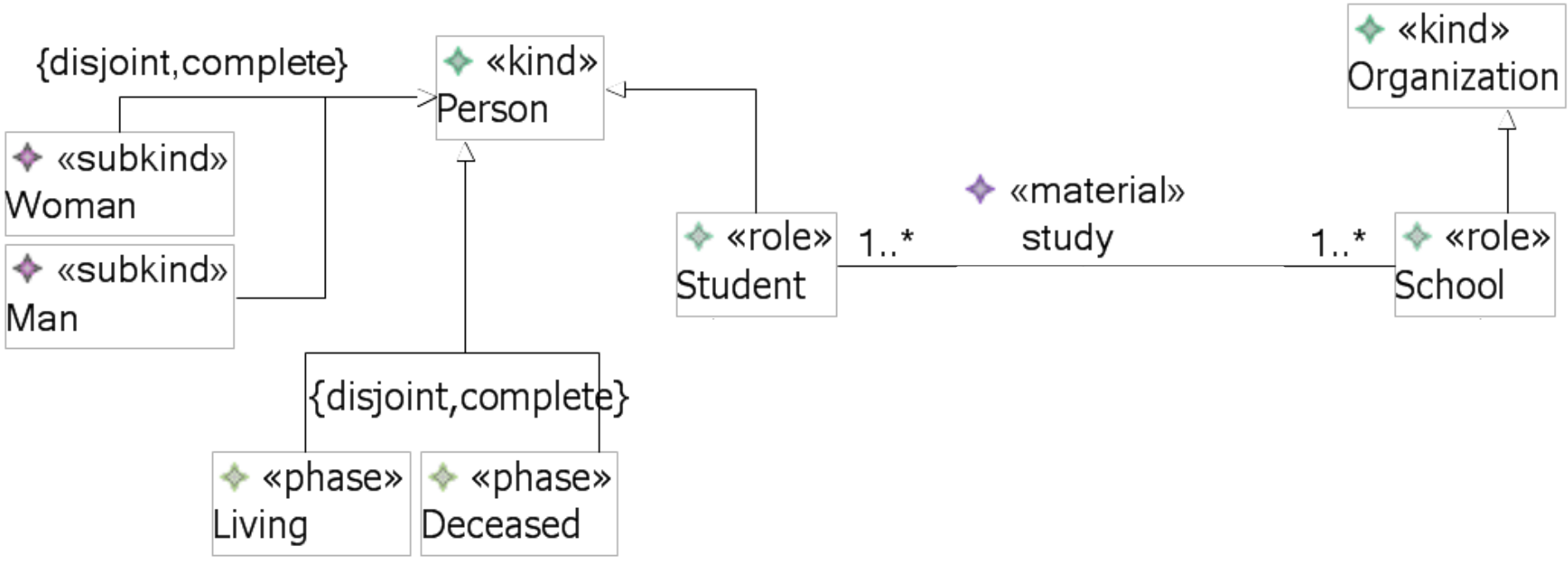
An Example



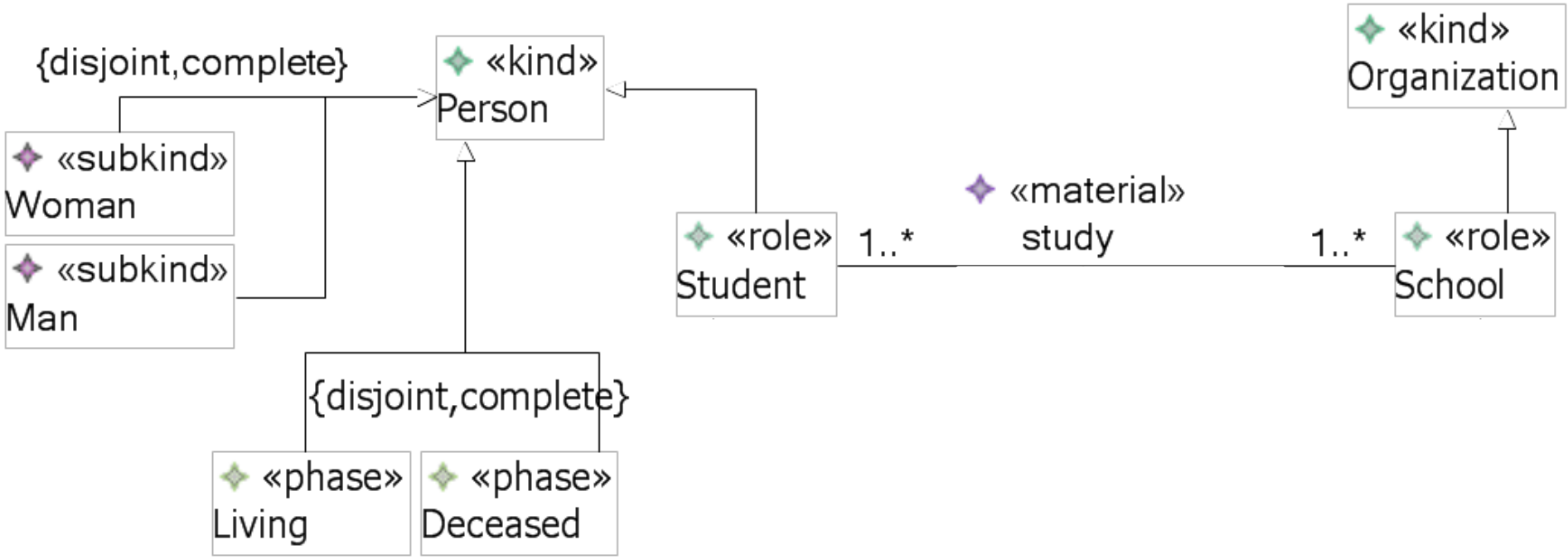
«kind»
Organization

$$\text{Anti Rigid}(T) \equiv \Box(\forall x(T(x) \rightarrow \Diamond(\neg T(x))))$$

An Example


{disjoint,complete}
«kind»
Person
«kind»
Organization
«subkind»
Woman
«subkind»
Man
«material»
study
«role»
Student
1..*
1..*
«role»
School
{disjoint,complete}
«phase»
Living
«phase»
Deceased

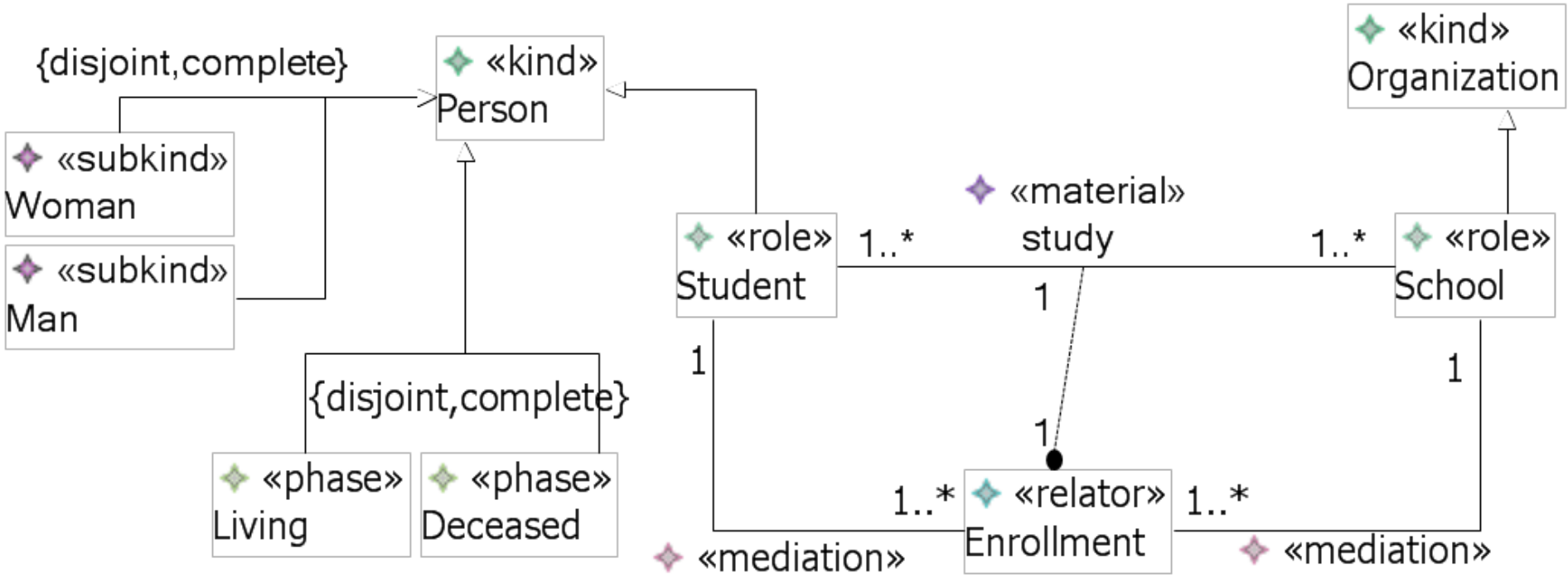
An Example



Relational dependency

$$D(T, P, R) \equiv \Box(\forall x(T(x) \rightarrow \exists y(P(y) \wedge R(x, y))))$$

An Example

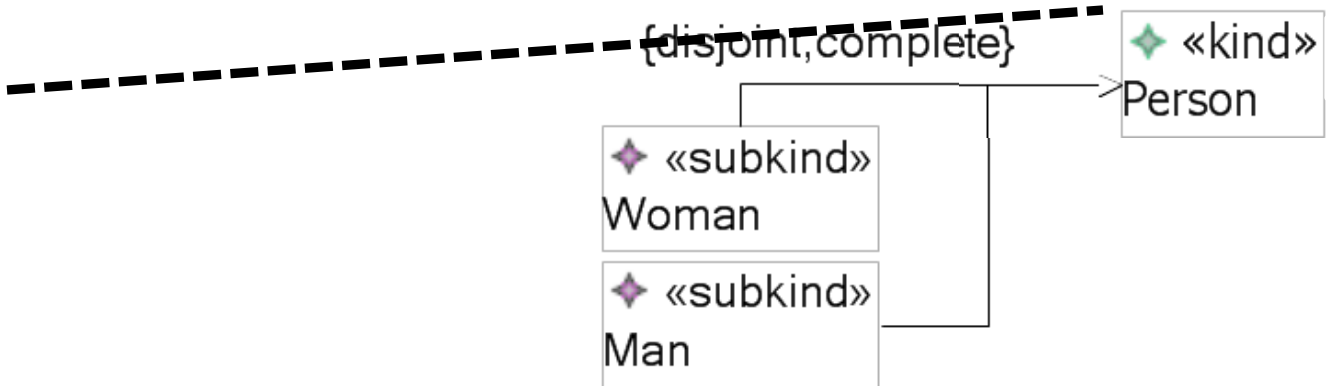

{disjoint,complete}
«kind»
Person
«kind»
Organization
«subkind»
Woman
«subkind»
Man
«material»
study
«role»
Student
1..*
1..*
«role»
School
1
1
1
1
{disjoint,complete}
«phase»
Living
«phase»
Deceased
1..*
«relator»
Enrollment
1..*
«mediation»
«mediation»

Alloy

- A model in Alloy consists of logical constraints which are captured in *signatures* and *fact declarations.*
- When a model is instantiated by the Alloy Analyzer, *atoms* are generated from signatures respecting the logical constraints in the model.
- Signatures can include field declarations, introducing relations between signatures.
- No notion of state change, dynamics or modality
- Analyzer can generate instances and produce counter-examples for predicates

Transformation

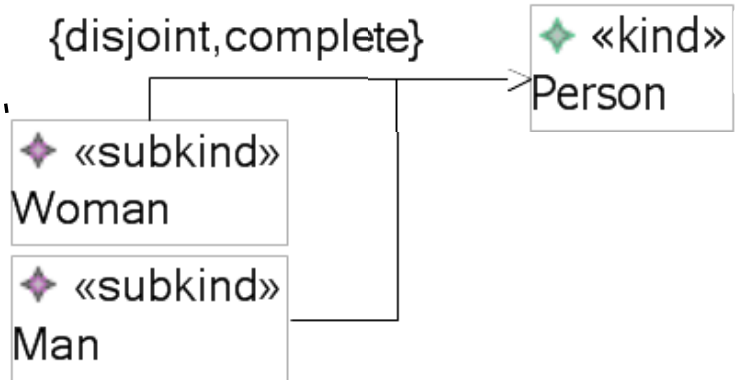
sig Person{}



Transformation

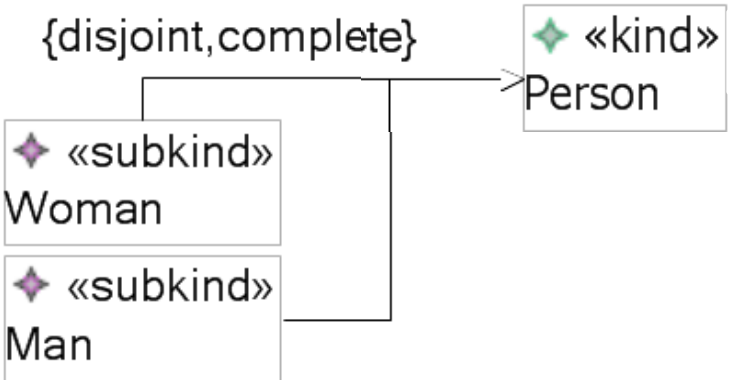
sig Person{}

sig Woman in Person{}



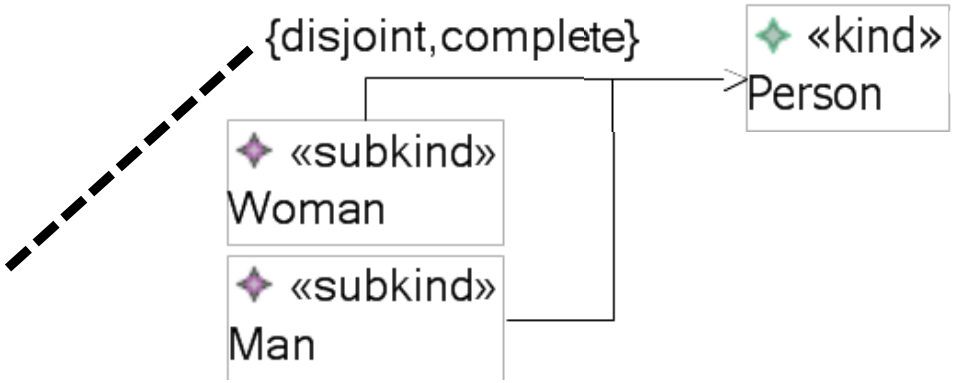
Transformation

sig Person{}
sig Woman in Person{}
sig Man in Person{}



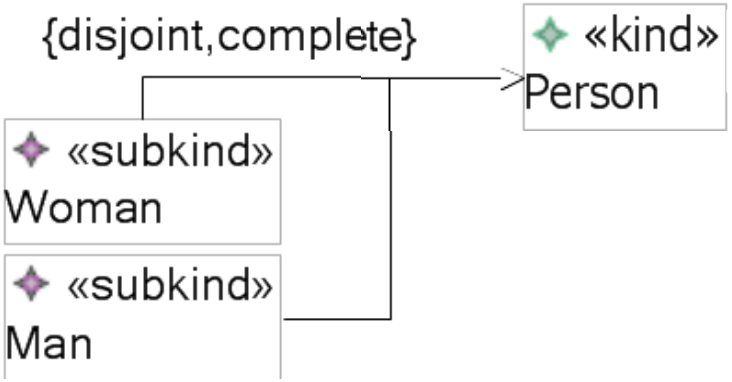
Transformation

```
sig Person{}
sig Woman in Person{}
sig Man in Person{}
fact{
Person = Woman+Man
disj[Woman,Man]
}
```



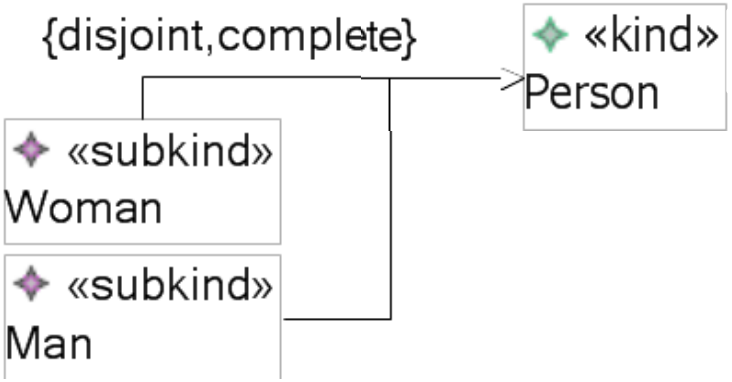
Transformation

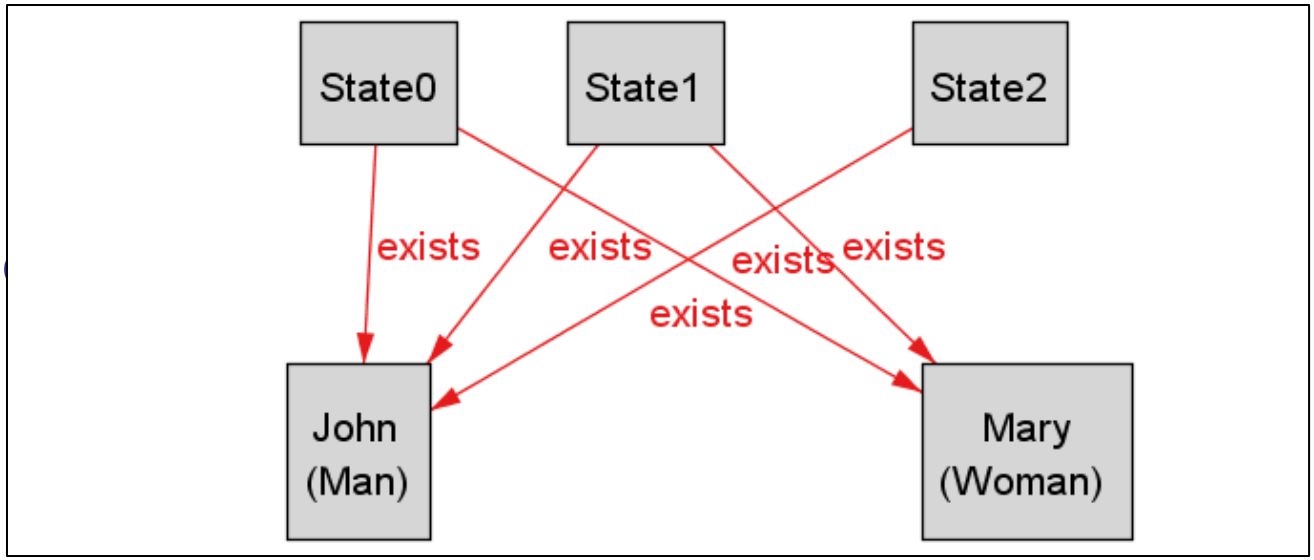
```
open util/ordering[State]
sig Person{}
sig Woman in Person{}
sig Man in Person{}
fact{
Person = Woman+Man
disj[Woman,Man]
}
sig State{
        exists: set Person,
}
```



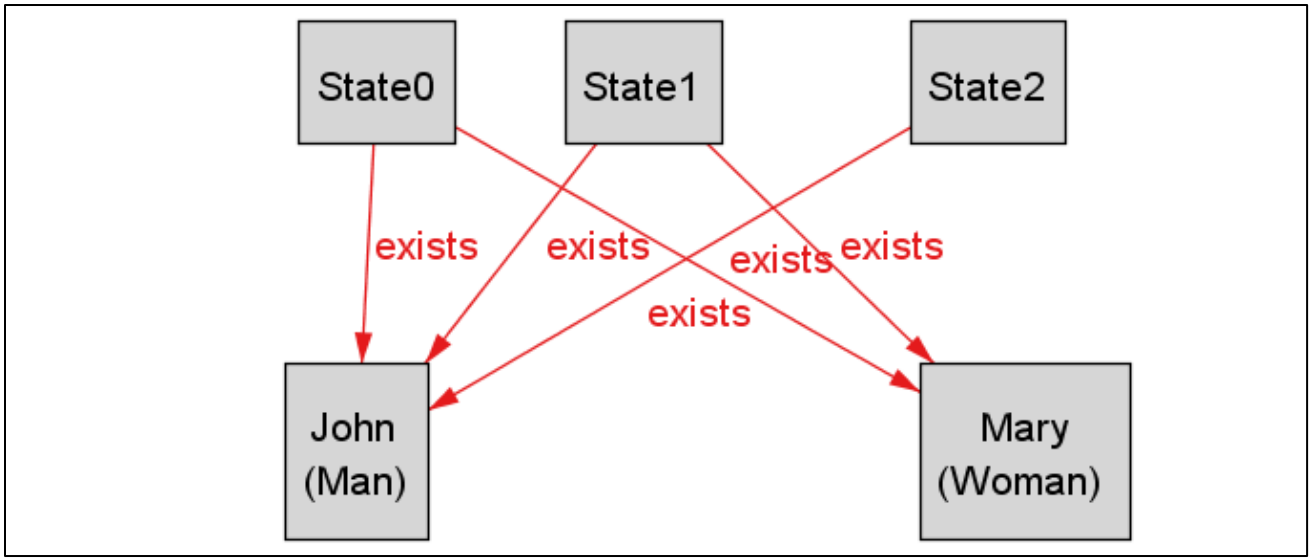
Transformation

```
open util/ordering[State]
sig Person{}
sig Woman in Person{}
sig Man in Person{}
fact{
Person = Woman+Man
disj[Woman,Man]
}
sig State{
      exists: set Pers
}
```

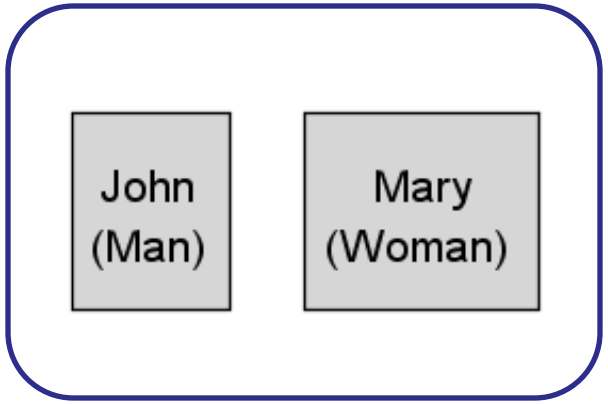




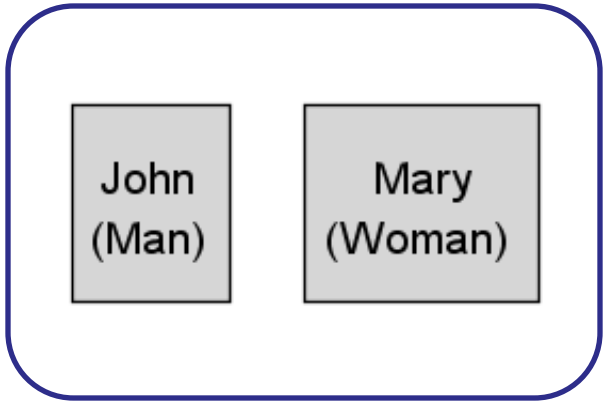
Transformation



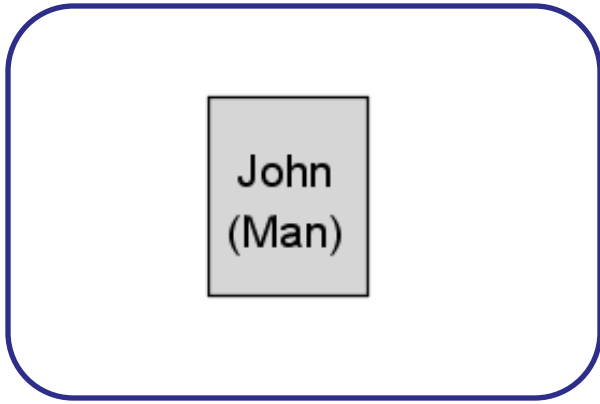
State 0



State 1

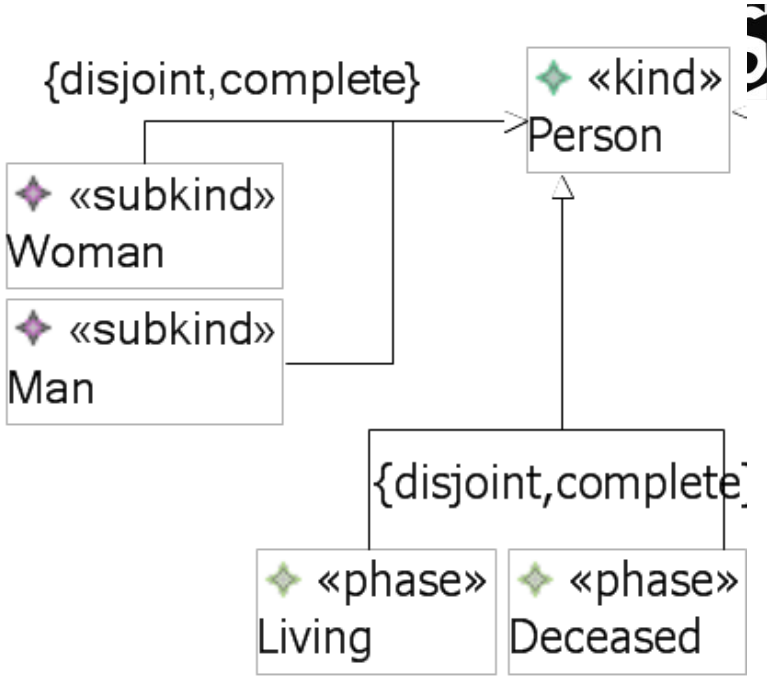


State2

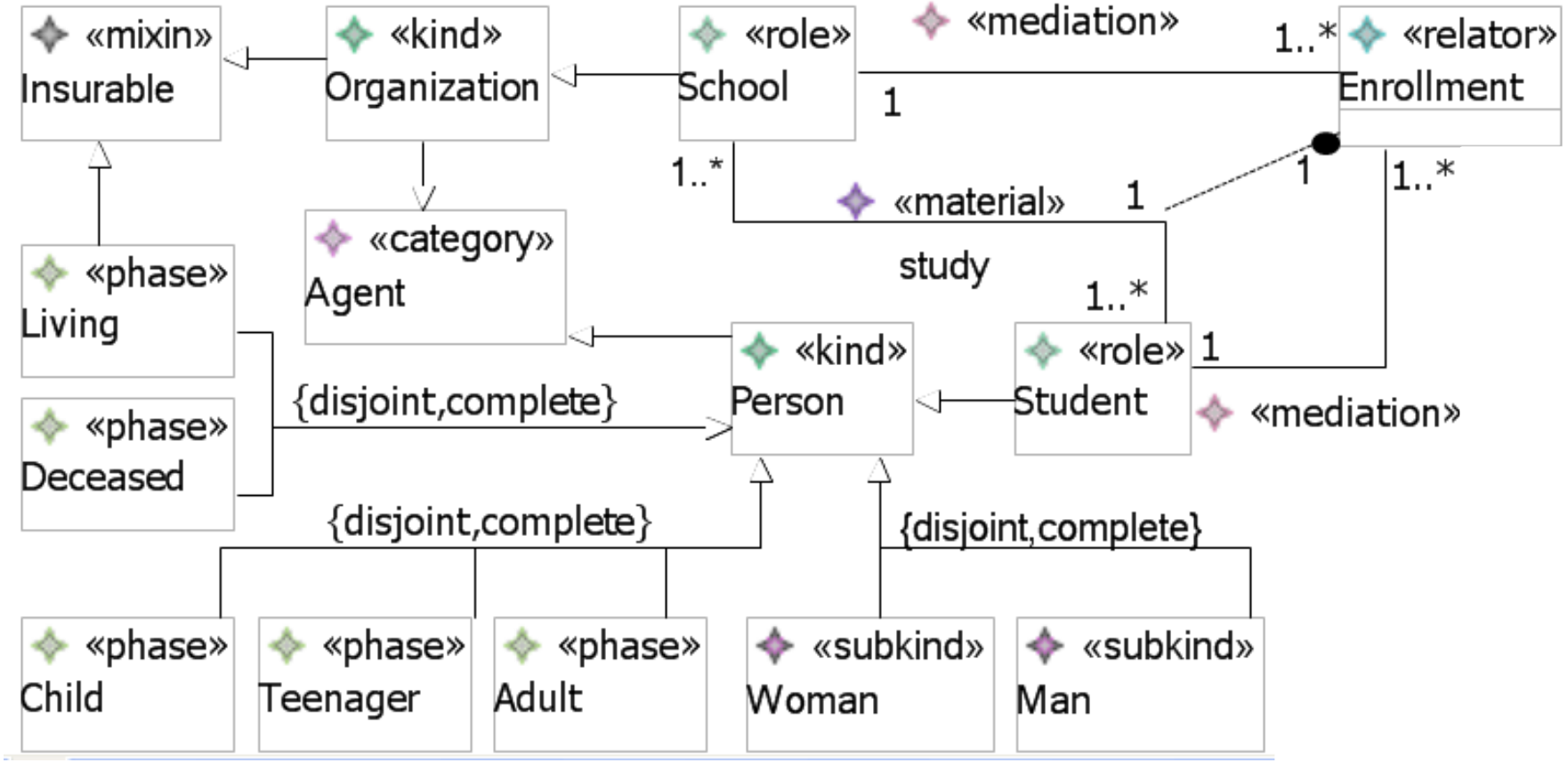


Transformation

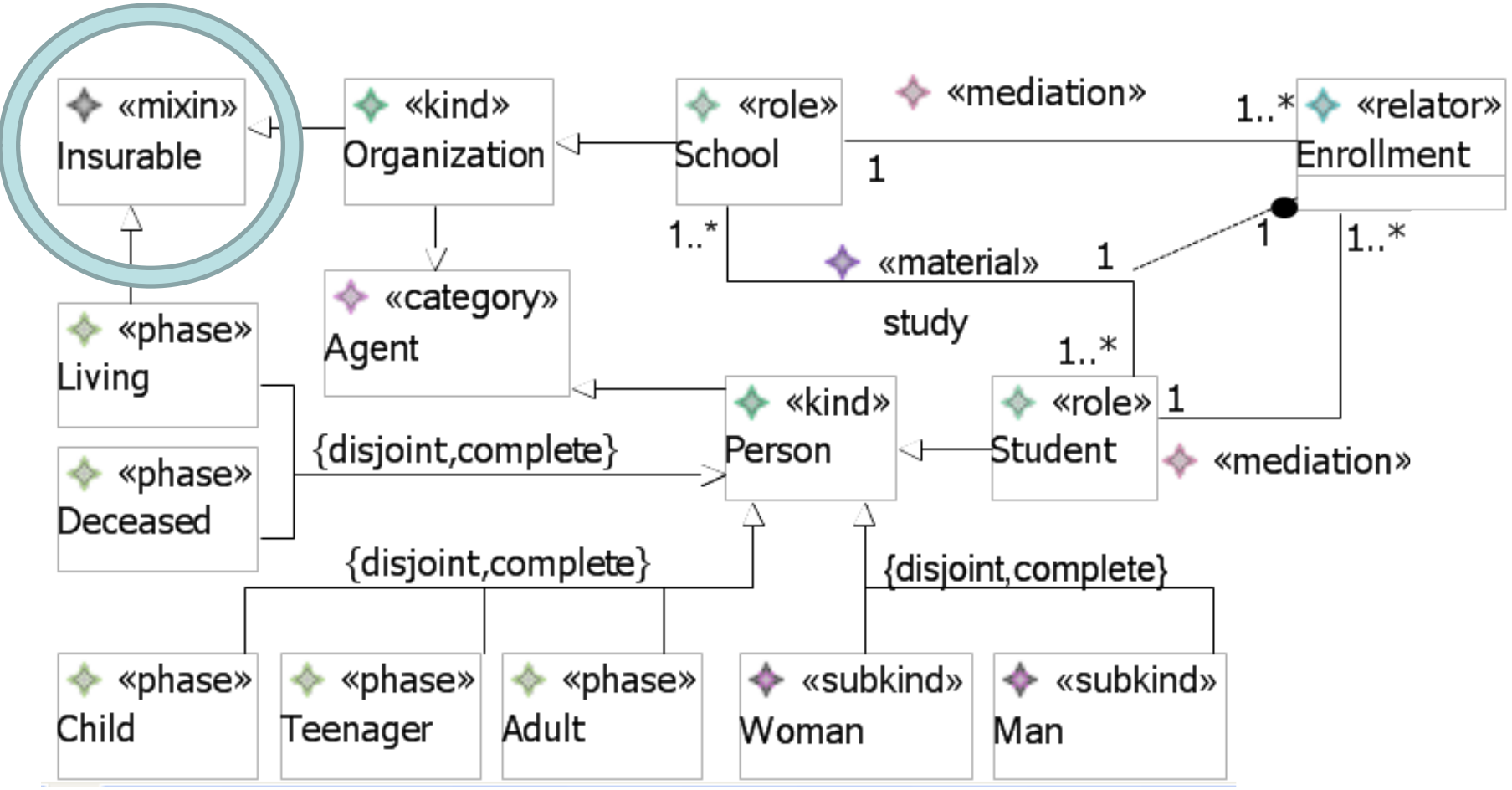
```
sig State{
        exists: set Person,
        disj Living, Deceased: exists
}
```



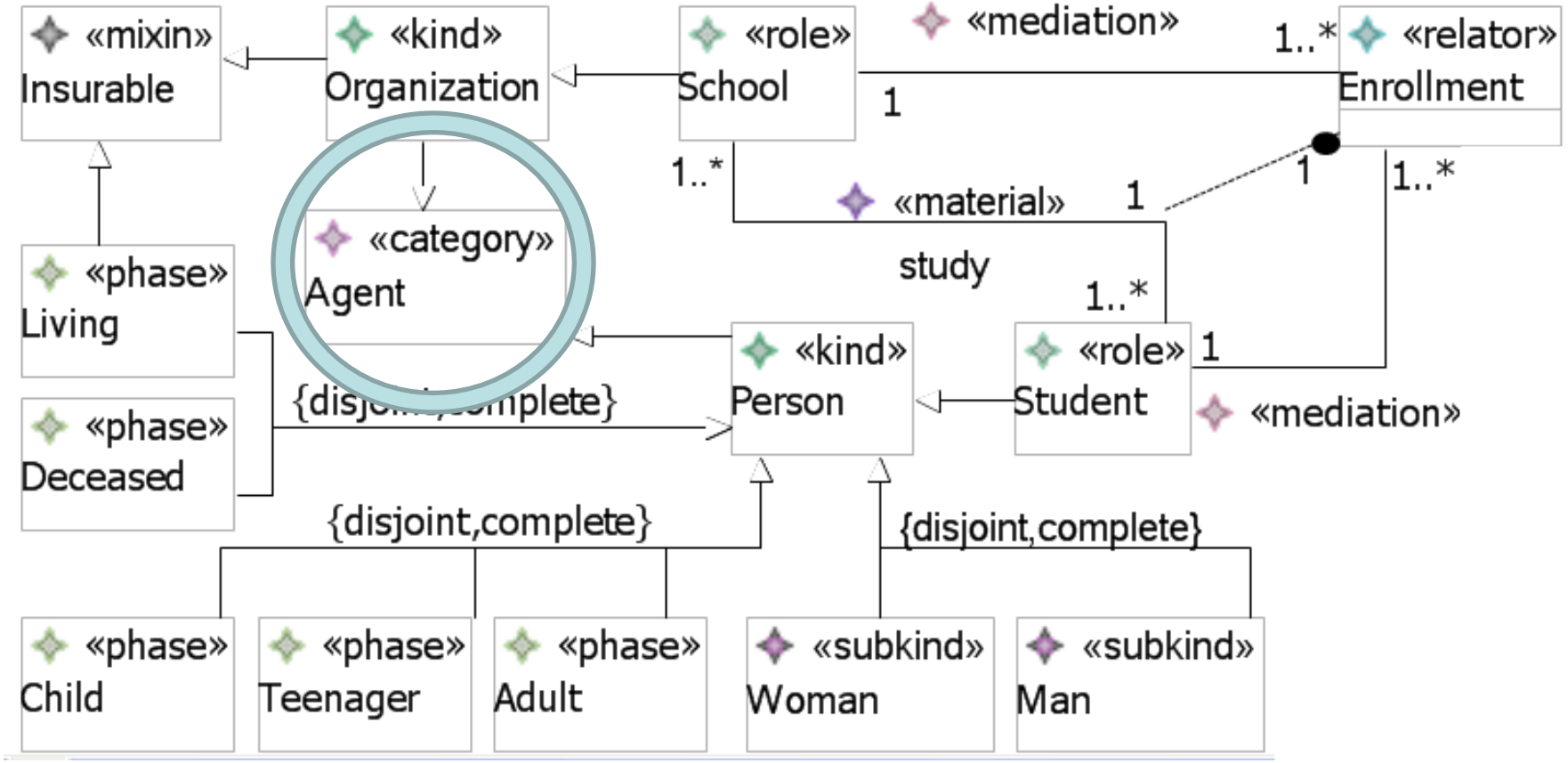
Example


«mixin»
Insurable
«kind»
Organization
«role»
School
«mediation»
1..*
«relator»
Enrollment
1
1..*
1
1
1..*
«material»
study
«category»
Agent
«phase»
Living
1..*
«kind»
Person
«role»
Student
1
«mediation»
«phase»
Deceased
{disjoint,complete}
{disjoint,complete}
{disjoint,complete}
«phase»
Child
«phase»
Teenager
«phase»
Adult
«subkind»
Woman
«subkind»
Man

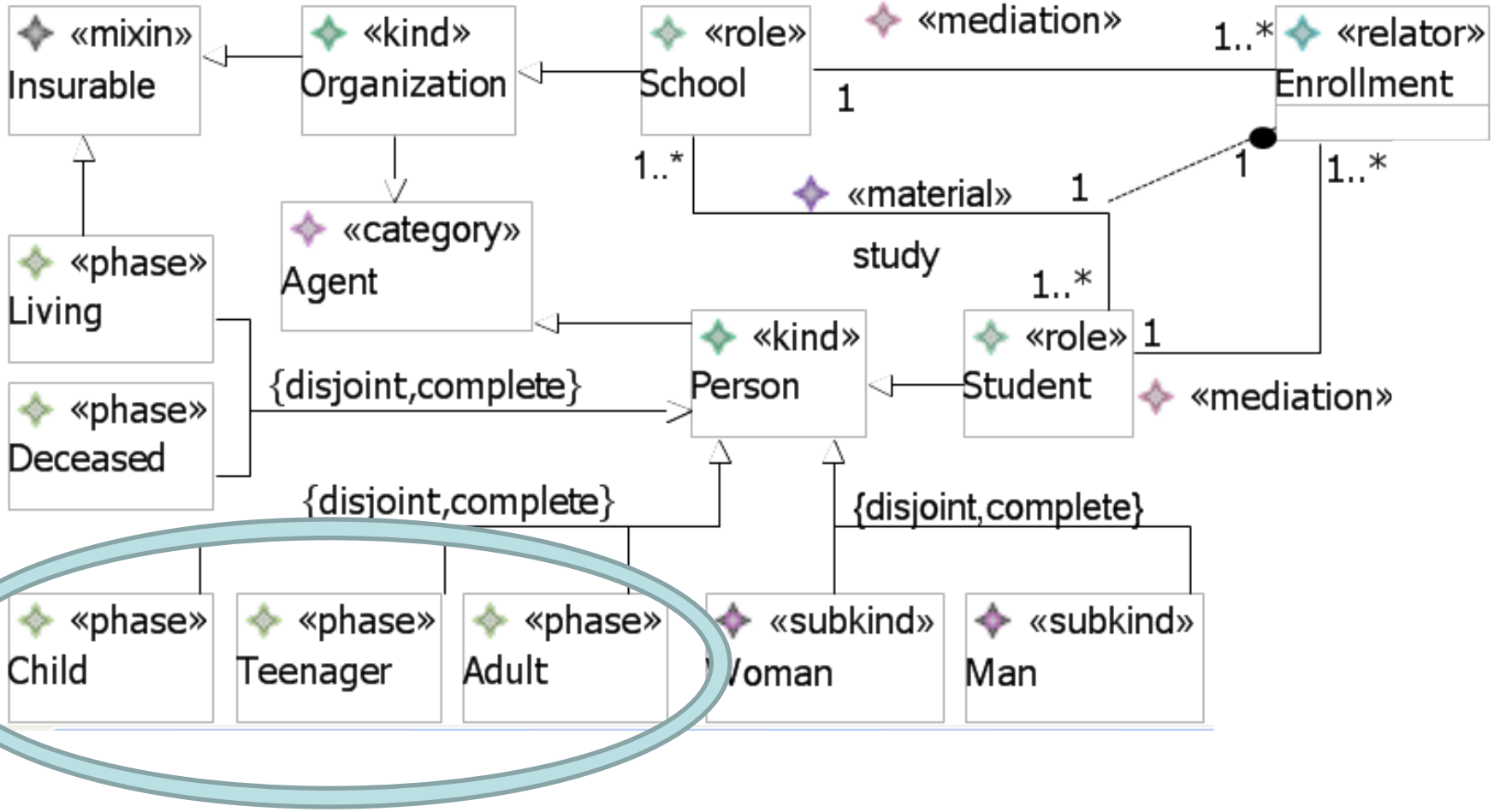
Example


«mixin»
Insurable
«kind»
Organization
«role»
School
«mediation»
1..*
«relator»
Enrollment
1
1..*
1
1..*
«material»
1
study
«category»
Agent
«phase»
Living
1..*
«kind»
Person
«role»
Student
1
«phase»
Deceased
{disjoint,complete}
«mediation»
{disjoint,complete}
{disjoint,complete}
«phase»
Child
«phase»
Teenager
«phase»
Adult
«subkind»
Woman
«subkind»
Man

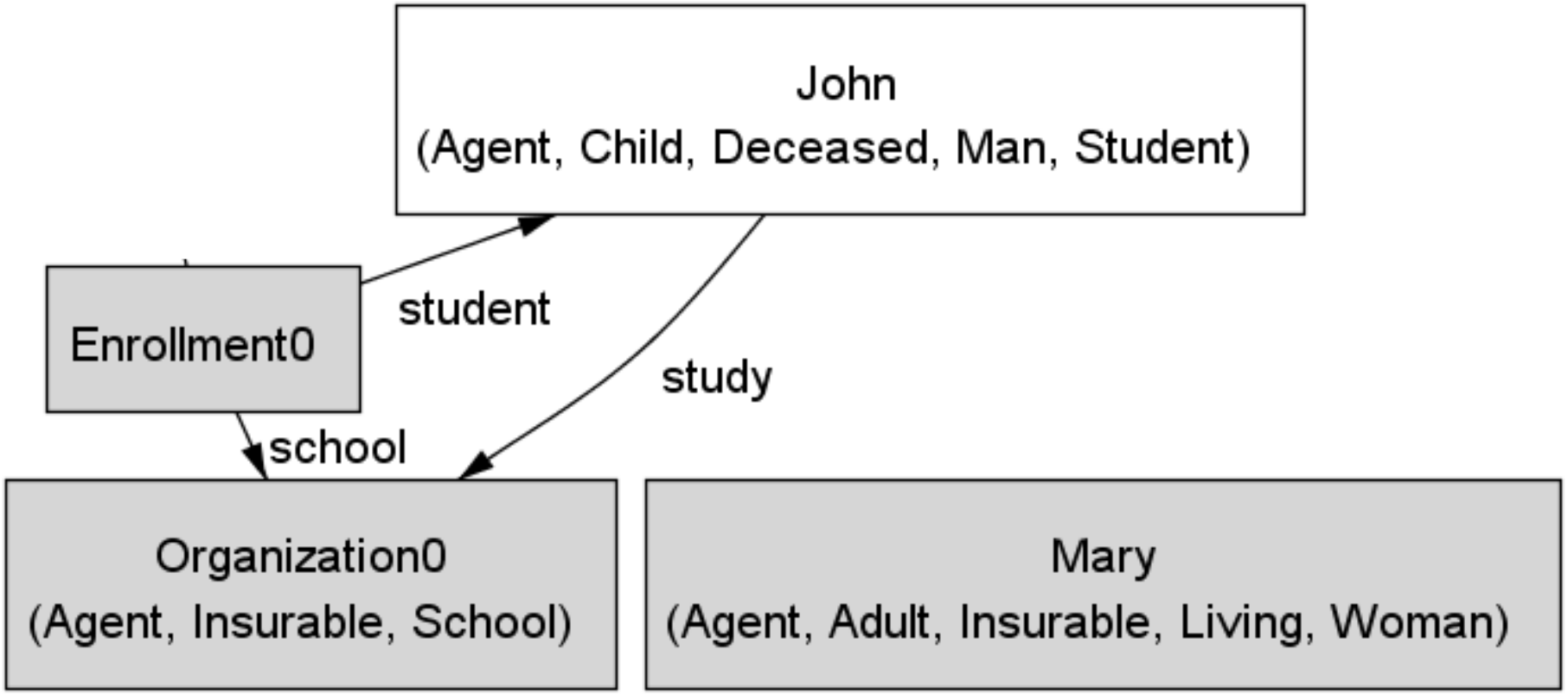
Example


«mixin»
Insurable
«kind»
Organization
«role»
School
«mediation»
1..*
«relator»
Enrollment
1
1..*
1
1
1..*
«material»
study
«category»
Agent
«phase»
Living
«phase»
Deceased
{disjoint,complete}
«kind»
Person
1..*
«role»
Student
1
«mediation»
{disjoint,complete}
{disjoint,complete}
«phase»
Child
«phase»
Teenager
«phase»
Adult
«subkind»
Woman
«subkind»
Man

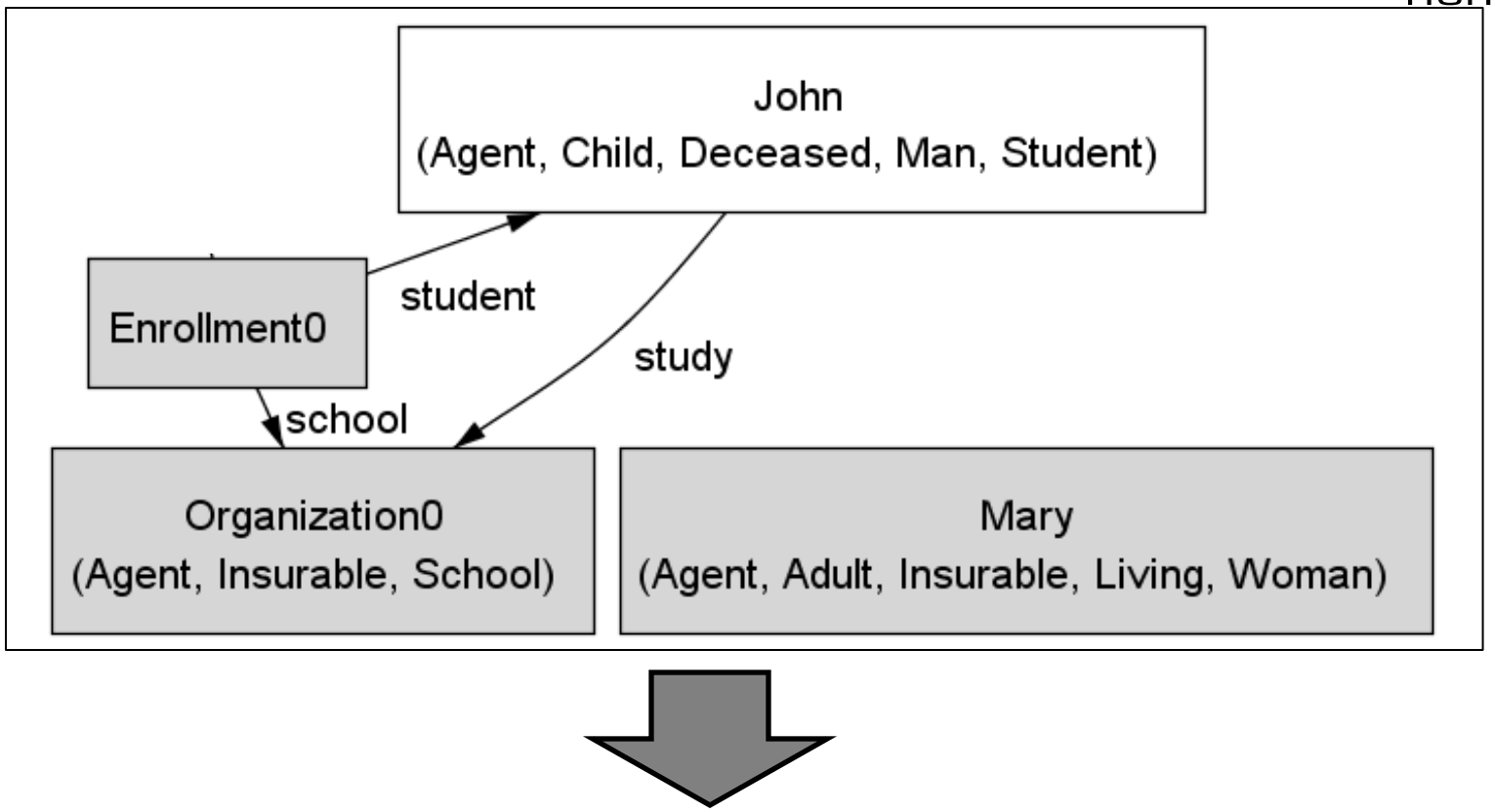
Example


«mixin»
Insurable
«kind»
Organization
«role»
School
«mediation»
1..*
«relator»
Enrollment
1
1..*
1
1..*
«material»
1
study
«category»
Agent
«phase»
Living
1..*
«kind»
Person
«role»
Student
1
«phase»
Deceased
{disjoint,complete}
«mediation»
{disjoint,complete}
{disjoint,complete}
«phase»
Child
«phase»
Teenager
«phase»
Adult
«subkind»
Woman
«subkind»
Man

Example

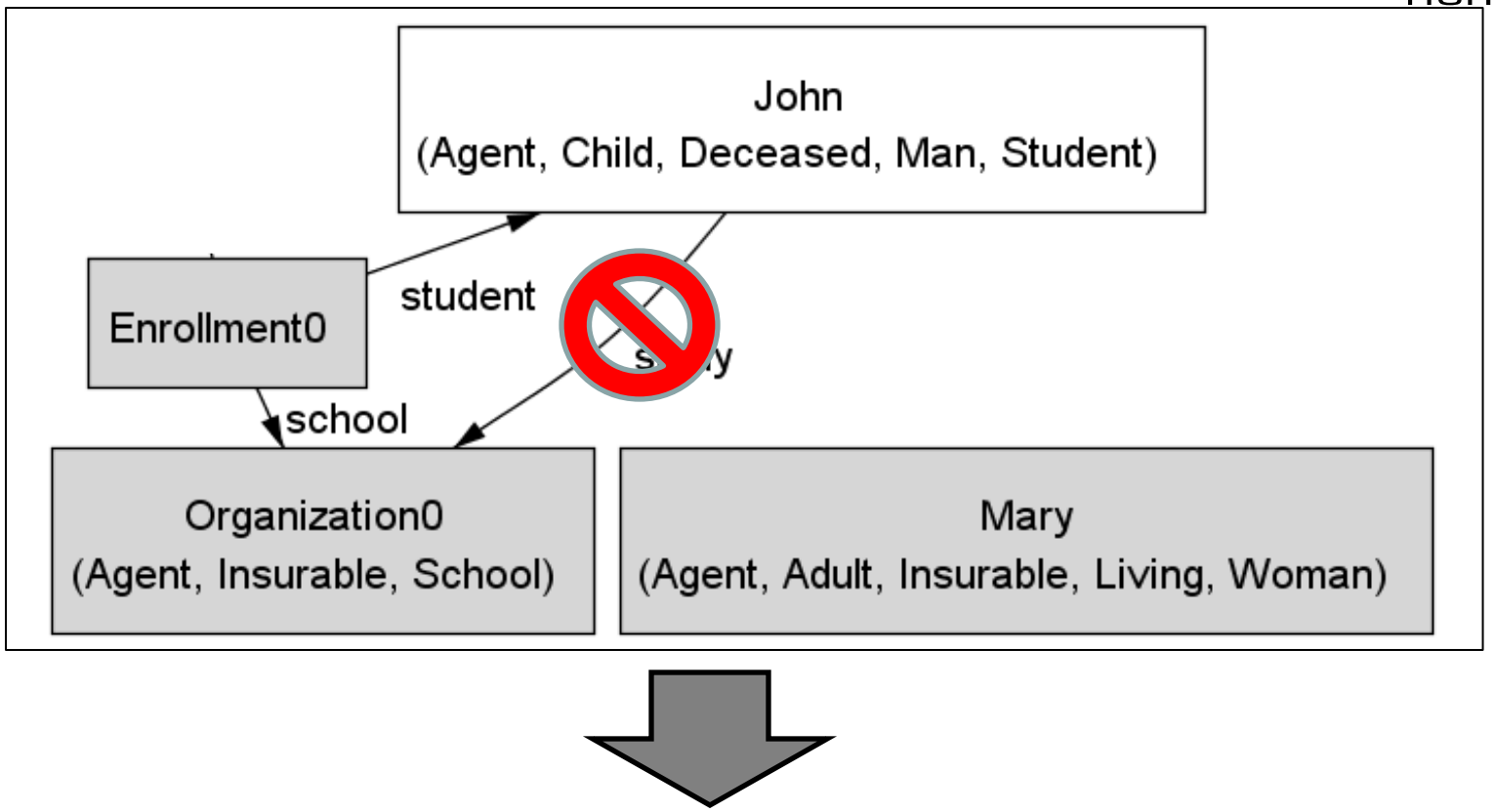

John
(Agent, Child, Deceased, Man, Student)
Enrollment0
student
study
school
Organization0
(Agent, Insurable, School)
Mary
(Agent, Adult, Insurable, Living, Woman)

Example



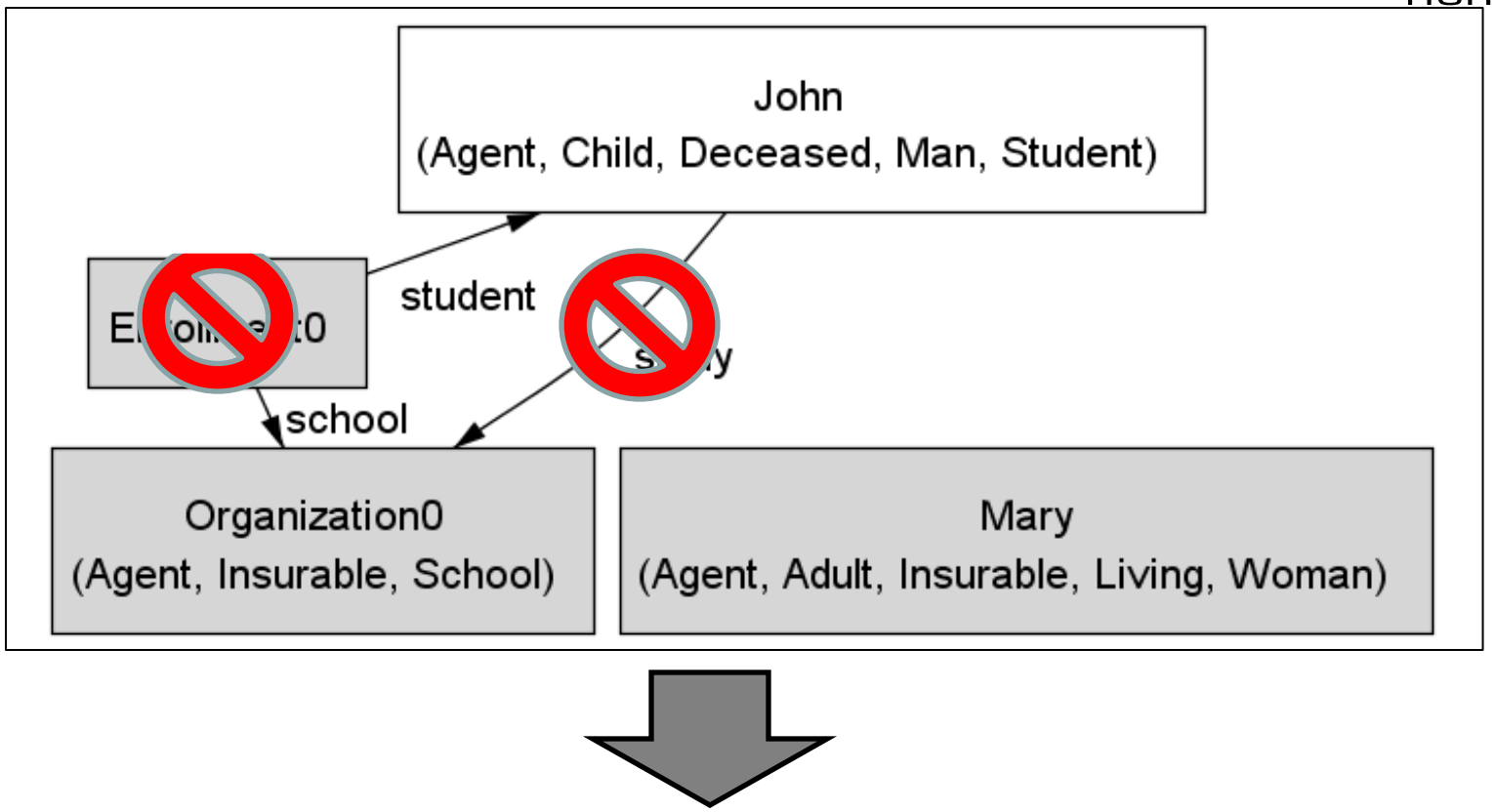
John	Mary	Organization0
(Agent, Child, Insurable, Living, Man)	(Agent, Insurable, Living, Teenager, Woman)	(Agent, Insurable)

Example



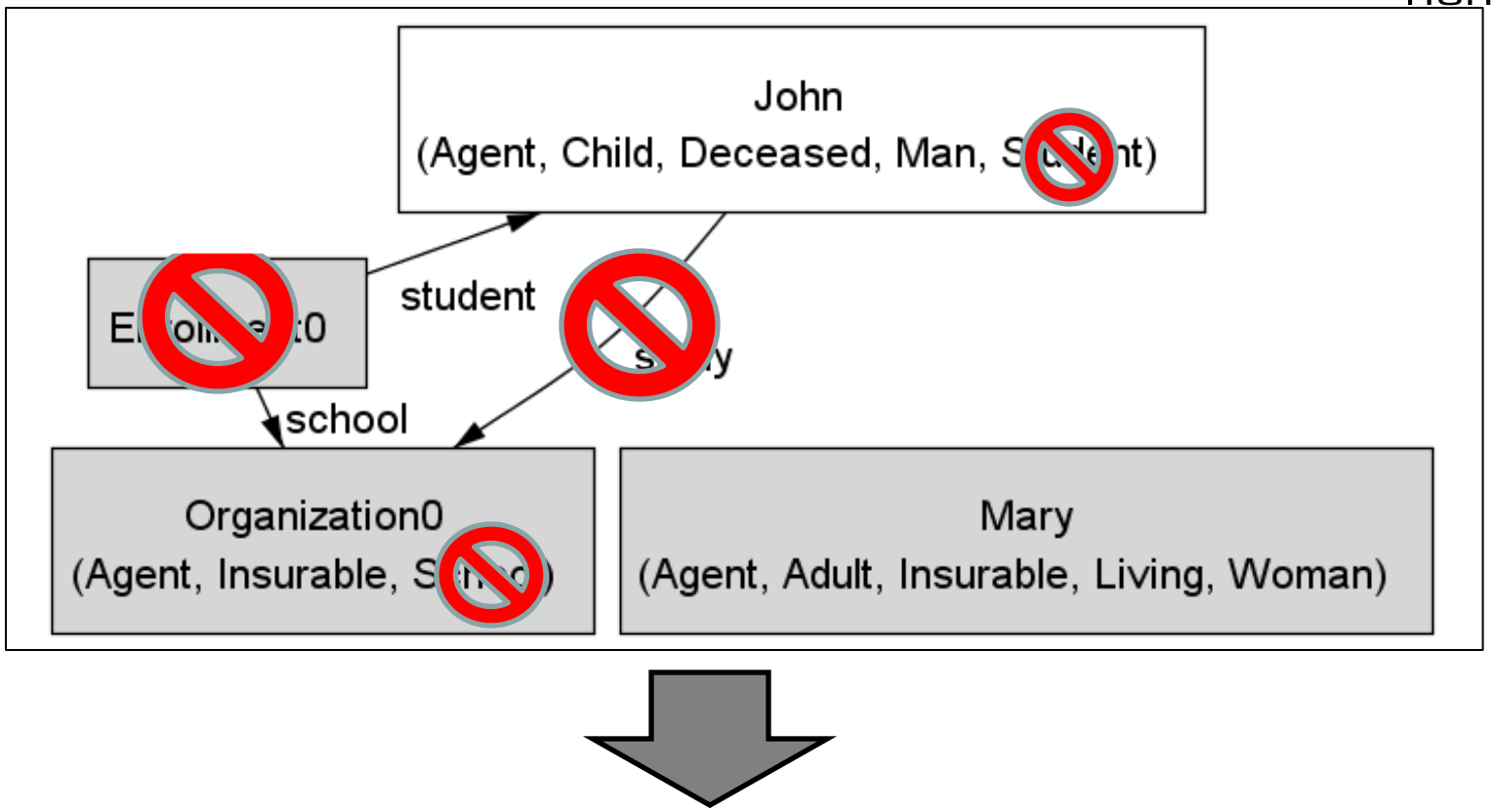
John	Mary	Organization0
(Agent, Child, Insurable, Living, Man)	(Agent, Insurable, Living, Teenager, Woman)	(Agent, Insurable)

Example



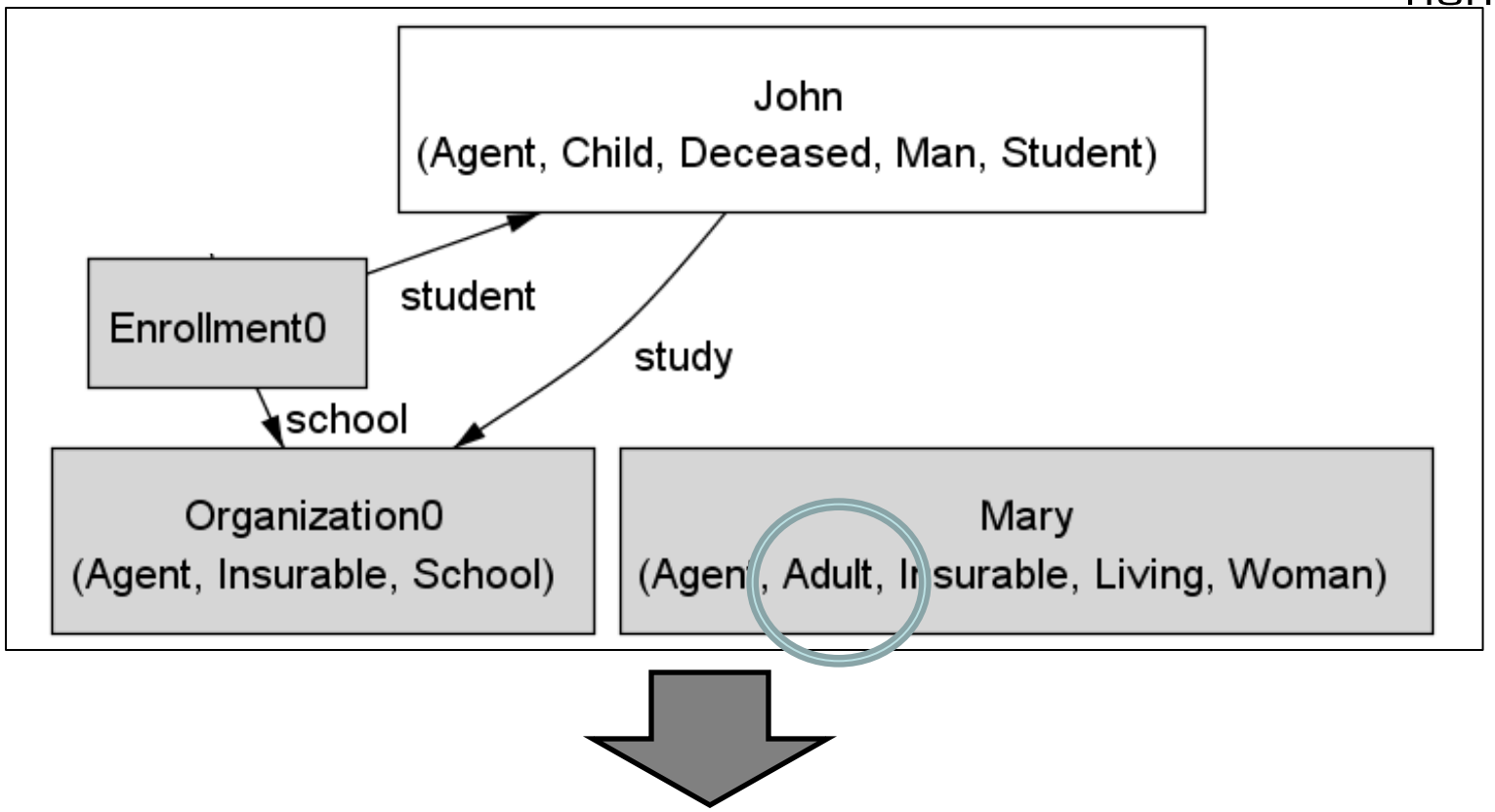
John	Mary	Organization0
(Agent, Child, Insurable, Living, Man)	(Agent, Insurable, Living, Teenager, Woman)	(Agent, Insurable)

Example



John (Agent, Child, Insurable, Living, Man)	Mary (Agent, Insurable, Living, Teenager, Woman)	Organization0 (Agent, Insurable)

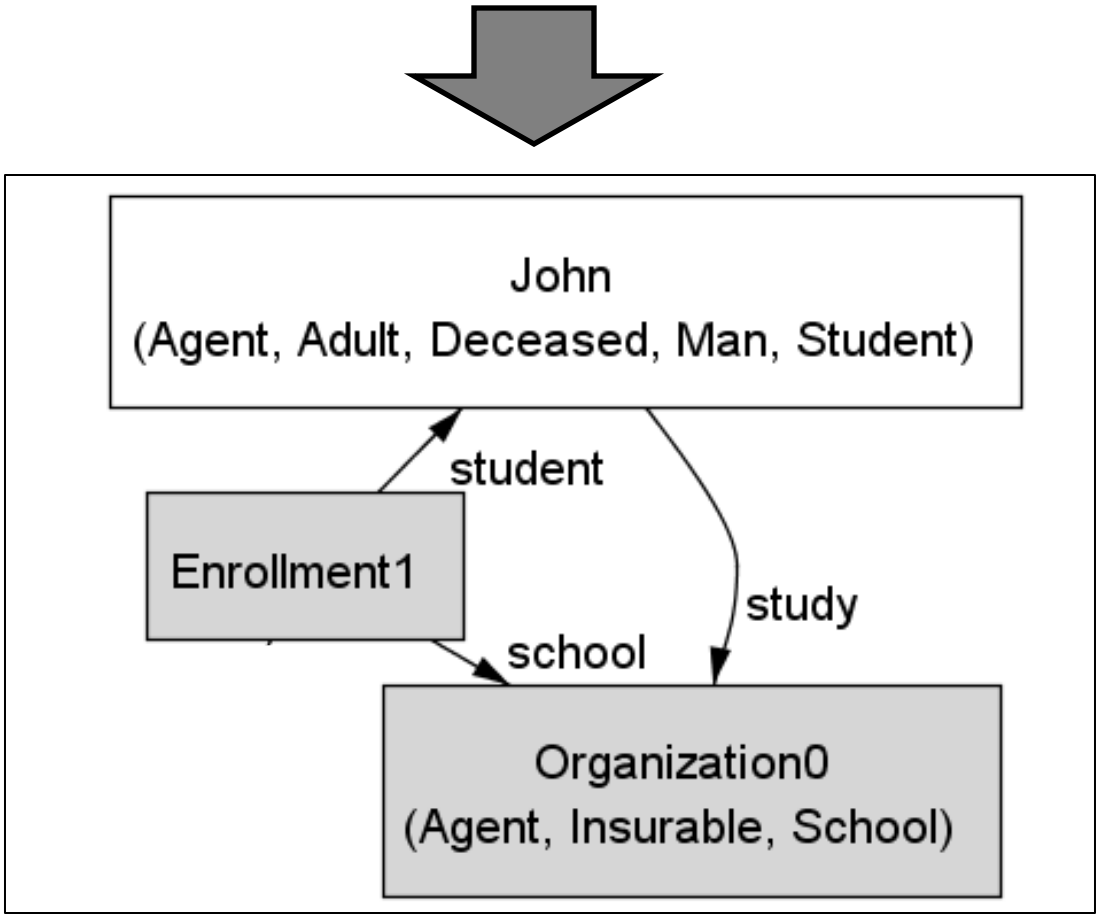
Example


John
(Agent, Child, Deceased, Man, Student)
Enrollment0
student
study
school
Organization0
(Agent, Insurable, School)
Mary
(Agent, Adult, Insurable, Living, Woman)


John
(Agent, Child, Insurable, Living, Man)
Mary
(Agent, Insurable, Living, Teenage, Woman)
Organization0
(Agent, Insurable)

Example

John (Agent, Child, Insurable, Living, Man)	Mary (Agent, Insurable, Living, Teenager, Woman)	Organization0 (Agent, Insurable)



Related Work

- Focus here on structural aspects of models
- USE tool (Gogolla et al. 2007)
  - Differently from our approach, which is based on the automatic creation of example state sequences, in USE the modeler must specify sequences of snapshots
- USE tool and Massoni et al. (2004) focus on single snapshots
- UML2Alloy (Anastasakis, Bordbar, et al. 2009) and Massoni et al. (2004) translate all classes into Alloy signatures, which suggest that no dynamic classification is possible in both approaches.
- None incorporate notions of modality

Conclusions

- A mature approach to conceptual modelling requires tools for modellers to gain confidence on the quality of the models they produce
- Our approach shows the possible dynamics of object creation, classification, association and destruction as defined in the conceptual model
- Snapshots confront the modeller with what he/she wrote

Future work

- Incorporate domain constraints / business rules (OCL->Alloy)
- Explore visualization techniques and perform empirical validation
- Methodological support for validation:
  - How can one select relevant scenarios?
- Assess scalability and performance
- Analysis and verification
  - What kinds of predicates are interesting for analysis?

About NEMO

http://nemo.inf.ufes.br

Ontology and Conceptual Modeling Research Group

Computer Science Department

Federal University of Espírito Santo

Av. Fernando Ferrari, s/n

29060-970 Vitória, ES, Brazil