Towards a novel model versioning approach based on the separation between linguistic and ontological aspects

Antonio Cicchetti, Federico Ciccozzi
Mälardalen University, IDT
Mail: {antonio.cicchetti, federico.ciccozzi}@mdh.se

Miami, 2013/09/30

Agenda

- Evolution management in MDE
  - Model differencing
  - Model merging
  - Metamodel evolution and model co-evolution
- Linguistic vs Ontological aspects
- A research agenda

Background

- Model-Driven Engineering (MDE) proposes to reduce complexity by shifting the focus from coding to modelling
  - From correctness-by-correction to correctness-by-construction
  - Analyses and code are the result of automated model transformation processes which are verified only once
  - The transformation processes guarantee that what is generated strictly corresponds to what is modelled

Background

- Model-Driven Engineering (MDE) proposes to reduce complexity by shifting the focus from coding to modelling
  - From correctness-by-correction to correctness-by-construction
  - Analyses and code are the result of automated model transformation processes which are verified only once
  - The transformation processes guarantee that what is generated strictly corresponds to what is modelled
- (Meta)models are subject of an evolutionary pressure comparable to the one typically faced by textual artefacts in code-centric approaches

Model differencing

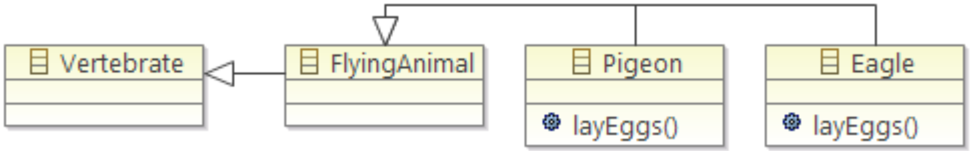
- Model manipulations must be detected at the same abstraction level of models
- Element matching relies on structural similarity and in general is a hard problem
- False-positives and negatives are unavoidable, in order to reach the desired level of precision user intervention is required

Model differencing

- Model manipulations must be detected at the same abstraction level of models

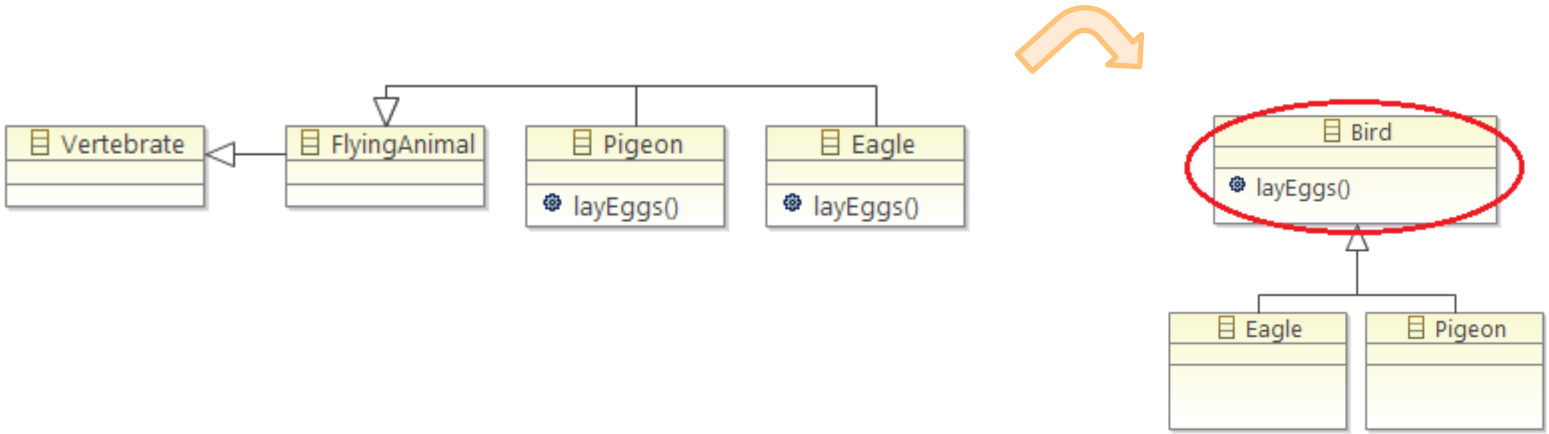
- Element matching relies on structural similarity and in general is a hard problem

- False-positives and negatives are unavoidable, in order to reach the desired level of precision user intervention is required



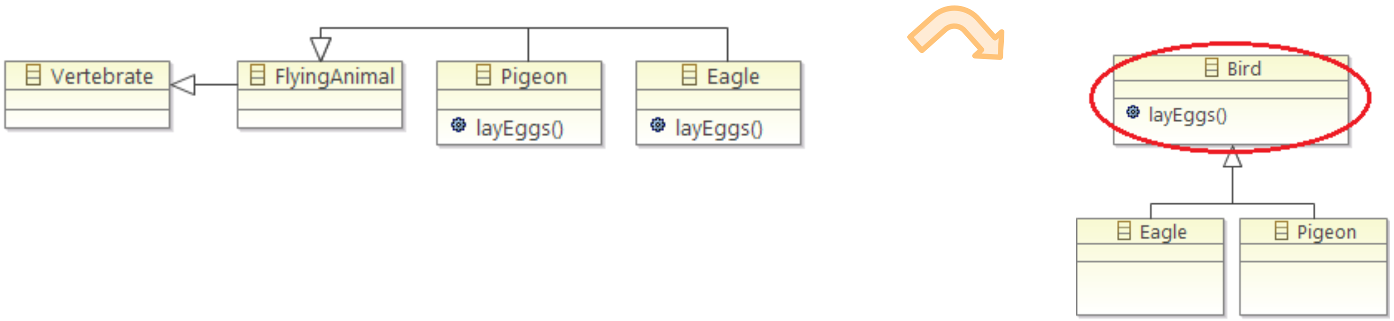
Model differencing

- Model manipulations must be detected at the same abstraction level of models
- Element matching relies on structural similarity and in general is a hard problem
- False-positives and negatives are unavoidable, in order to reach the desired level of precision user intervention is required



Metamodel evolution and model co-evolution

- When a metamodel evolves conformance of existing models can be broken
- Co-evolution countermeasures re-establish model conformance based on the metamodel evolution
- If metamodel evolution is erroneously detected migration strategies can cause loss of data and (semantically) incorrect models
- In general arbitrary precision requires users to specify explicitly their own migration strategies



Model merging

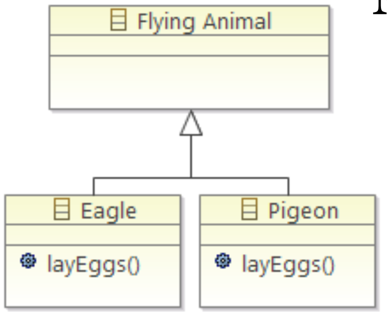
- Model merging is needed whenever concurrent changes are made on (portions of) the same model

- Distributed and multiview modelling are two scenarios that need model merging

- Merging models require an appropriate conflict detection and resolution support

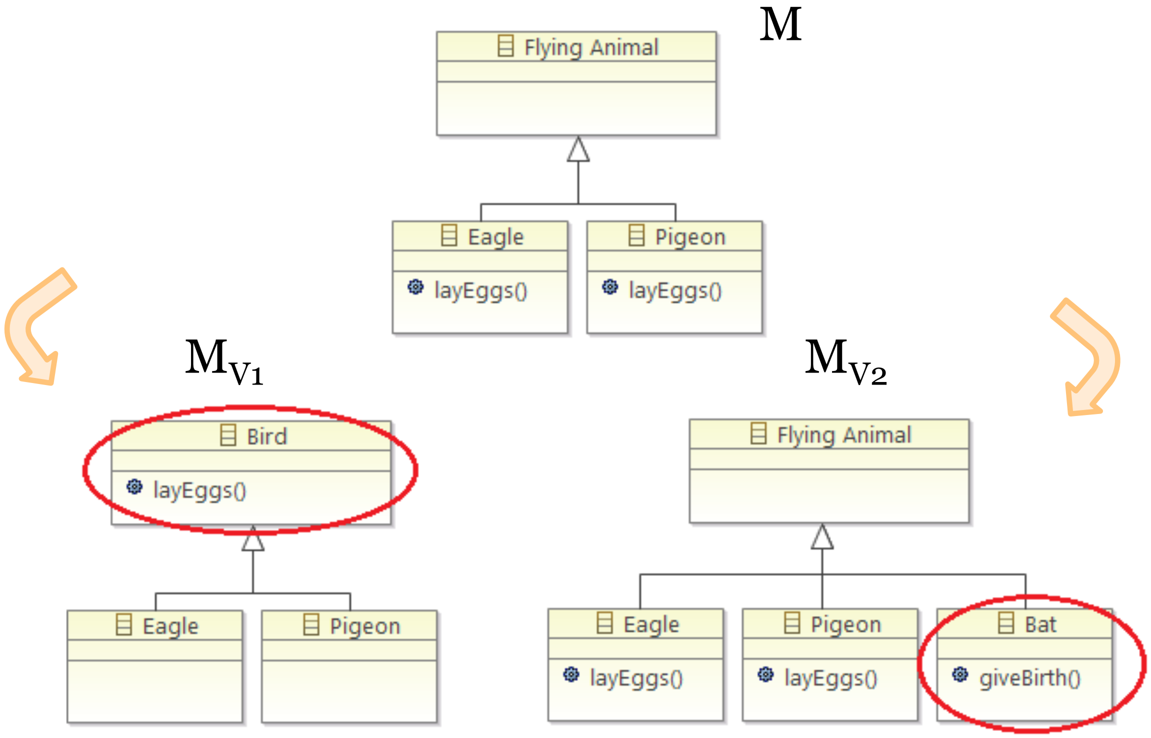
- When semantics comes into account, arbitrary precision demands the user to explicitly define conflict problems

Model merging

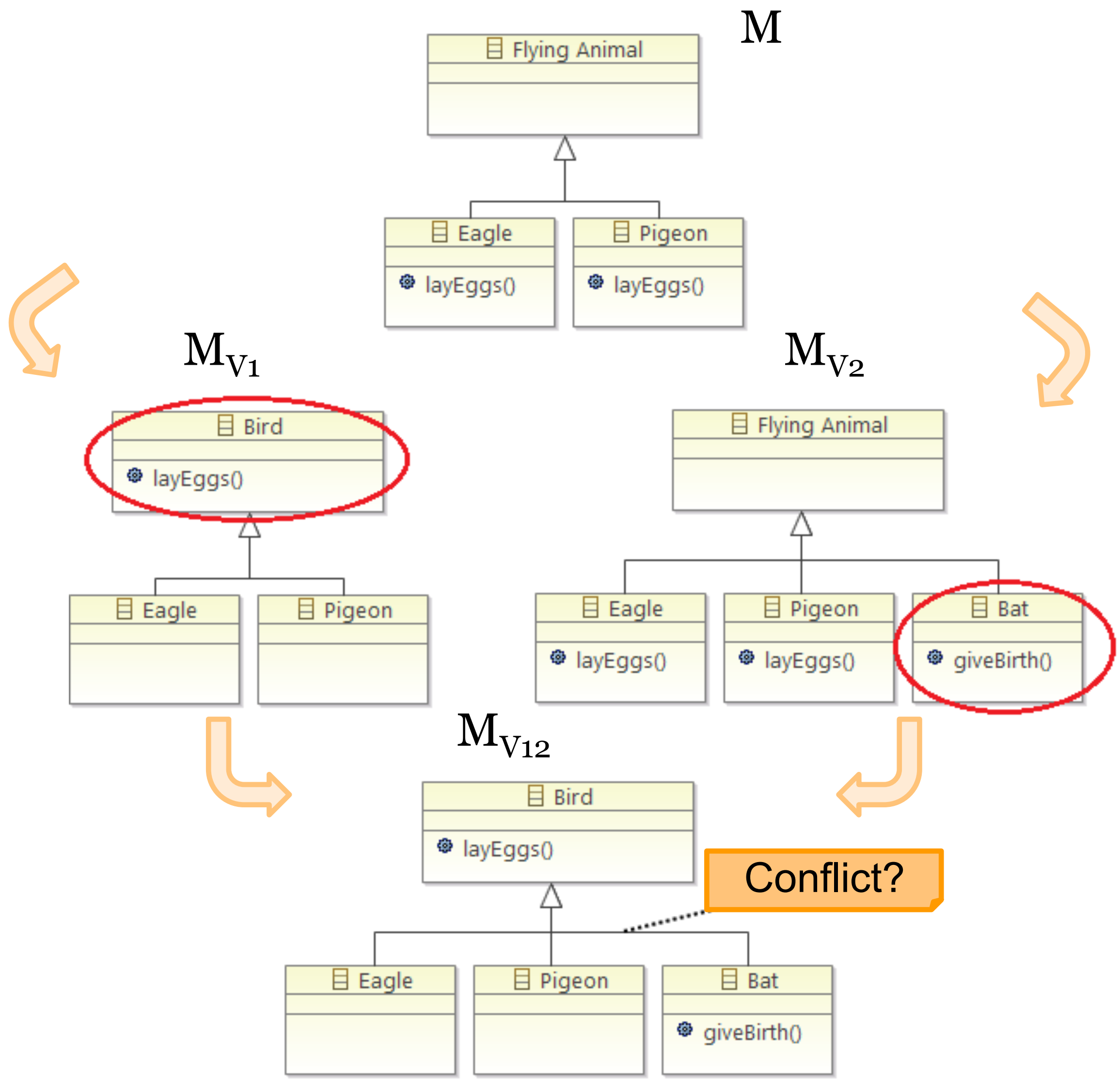


M

Model merging



Model merging

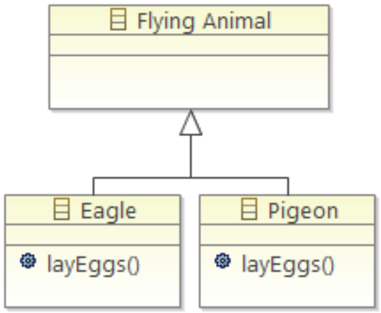


Linguistic vs Ontological aspects

- (Meta-)Modelling activities carry along not only the syntax by which concepts are expressed but also the underlying semantics of the application domain
- Syntax and semantics can be intertwined due to the adopted syntax and structural constraints
- Kühne proposed an alternative separation between those two aspects by introducing linguistic and ontological matters of (meta-)modelling [1]
- Linguistic aspects are constraints and rules that define the structural correctness of a model
- Ontological aspects are those pertaining to the domain taken into account, and exploit structural compositions to prescribe domain-specific well-formedness
- Therefore they create a new logical abstraction level by specializing (groups of) concepts at lower levels of abstraction

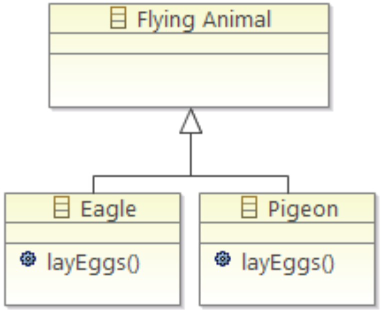
[1] Thomas Kühne: Matters of (Meta-)Modeling. Software and System Modeling 5(4): 369-385 (2006)

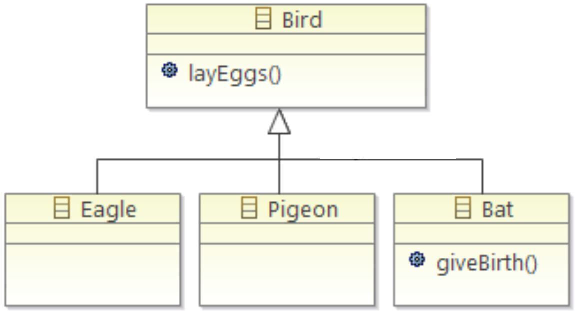
Linguistic vs Ontological aspects



- From a linguistic point of view, we have two classes that specialize a common superclass and each of the specializations owns an operation

Linguistic vs Ontological aspects

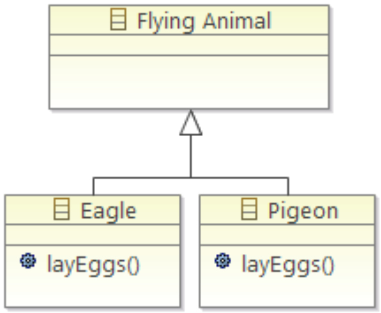


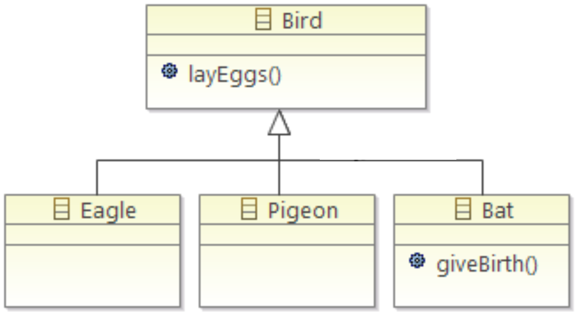


- From a linguistic point of view, we have two classes that specialize a common superclass and each of the specializations owns an operation

- In the same way, *Bird* is a superclass owning an operation which is inherited by its three specializations, *Eagle*, *Pigeon*, and *Bat*

Linguistic vs Ontological aspects





- From an ontological point of view things change: if we adopt an animal classification ontology the model on the right is not correct (at least Bat does not lay eggs)

- It is worth noting that different ontologies can give place to different domain-specific issues (for an OO programming language ontology both the models are still correct)

Linguistic vs Ontological aspects

- Linguistic and ontological separation has been already exploited to relax model conformance problems in the case of metamodel evolution

- Other approaches leverage on this separation to provide generic transformation strategies (based on linguistic aspects) which are later on bound to domain-specific concepts (i.e. ontological aspects)

Our proposal

- We propose to leverage on the decoupling between linguistic and ontological aspects to improve evolution support in MDE

- While there already exists a number of works able to deal with semantics issues, the semantics gets coupled (hidden) with the solution adopted for each specific case

- Our aim is to reduce the complexity of evolution support by addressing linguistic and ontological points of view in a separate manner

- An interesting general advantage of such a separation is to keep solved linguistic issues while leaving variability in modelling ontological aspects

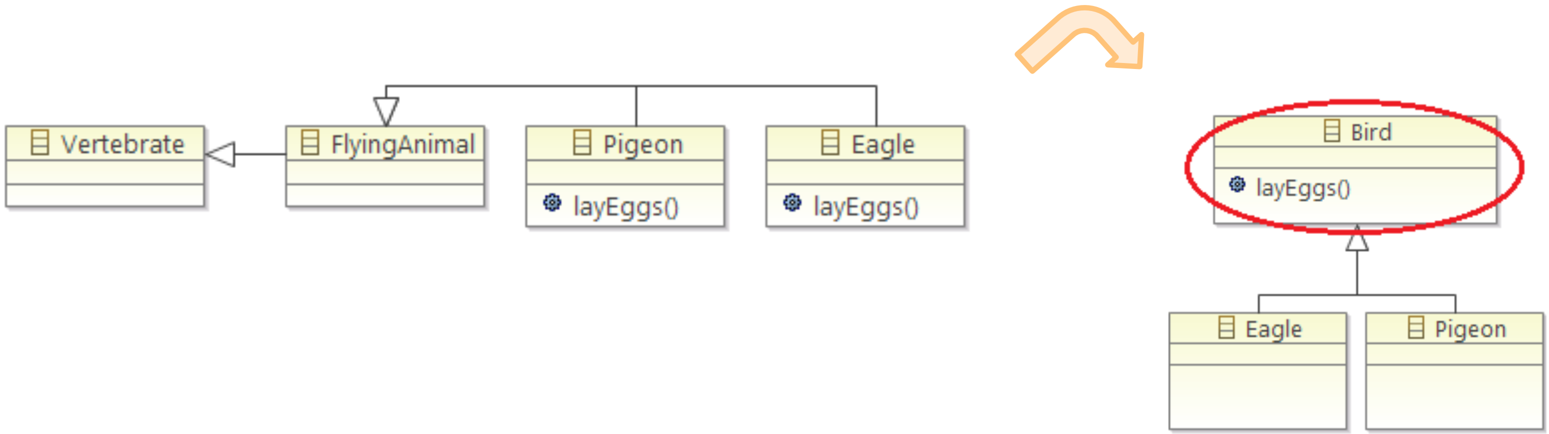
- As a consequence, we propose to revisit current MDE evolution techniques taking into account such aspects

Model versioning

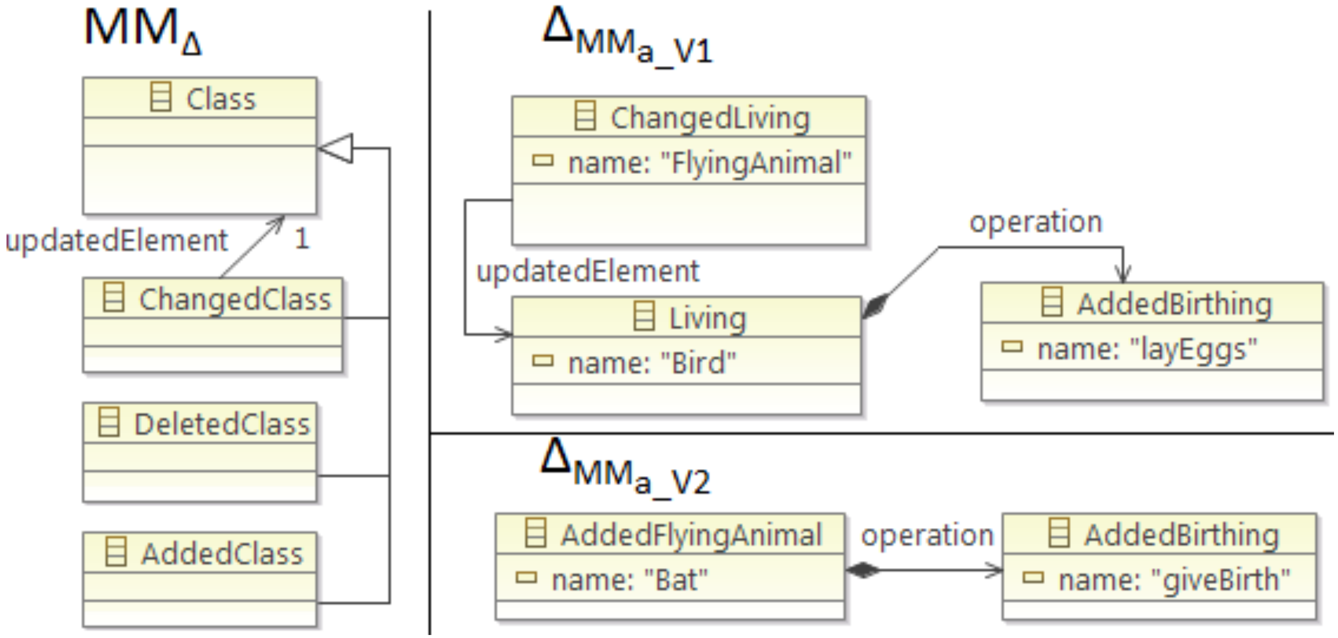
- Differencing can be performed from a linguistic or an ontological perspective, or a combination of both

Model versioning

- Differencing can be performed from a linguistic or an ontological perspective, or a combination of both
- If we take into account both linguistic and ontology, *FlyingAnimal* has been renamed to Bird and an *ExtractOperation* refactoring operation has been applied to move *layEggs* to the parent class



Model versioning



Metamodel evolution and model coevolution

- Manipulation detection for metamodels can be done exploiting the same reasoning done for models, since metamodels are models themselves
- Linguistic evolutions are typically easier to manage in terms of coevolution strategies
- However, it should be noticed that some linguistic manipulations could result in ontological evolutions
- Ontological aspects can help in driving coevolution strategies and in identifying erroneous alternatives (e.g. when customized by the user)
- Furthermore, in general ontological aspects have side effects on the metamodel ecosystem, notably transformations, editors, and so forth

Model merging

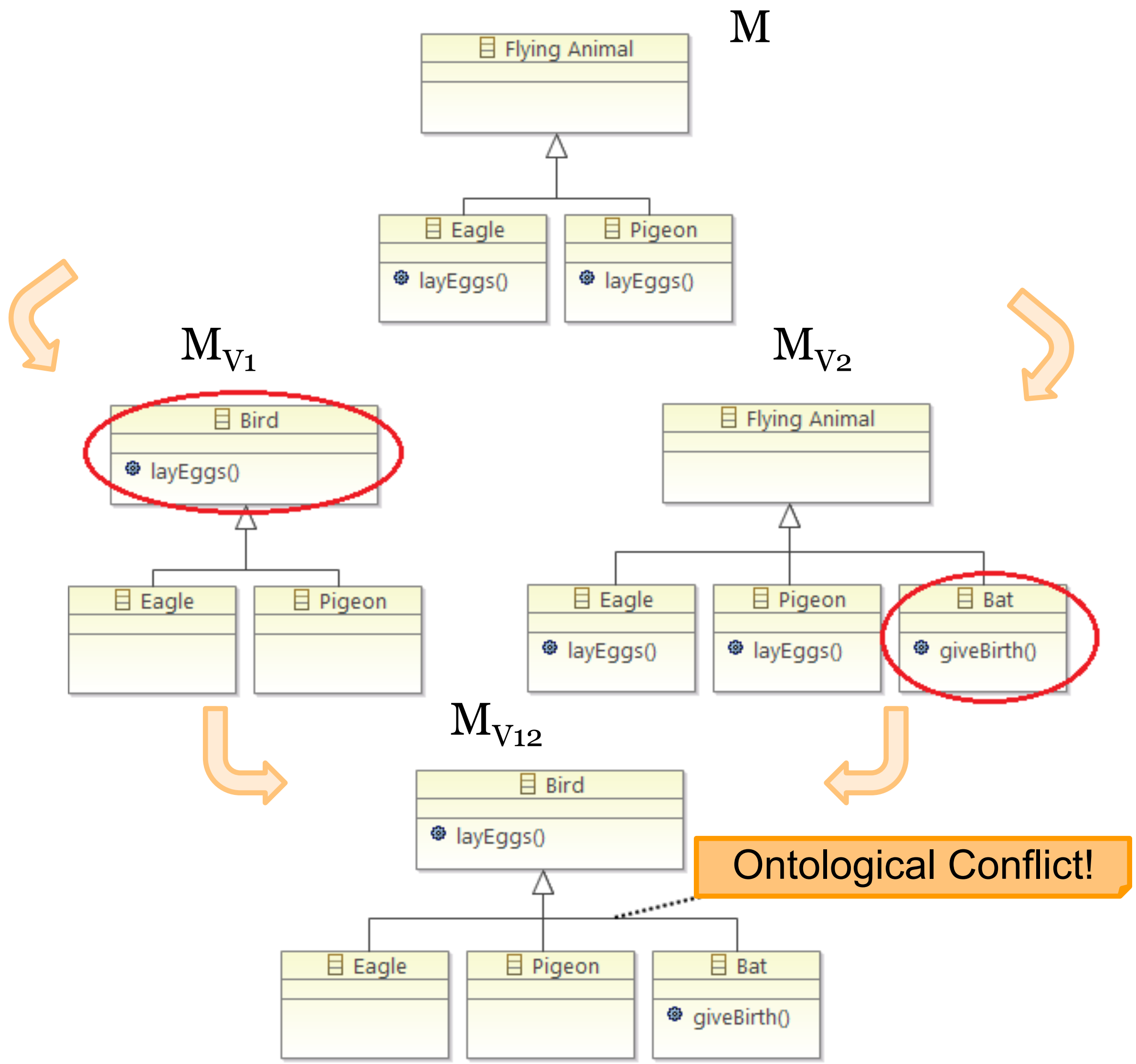
- Conflict detection and resolution can be tackled by distinguishing linguistic and ontological aspects

- Linguistic aspects have to be fixed, since these kind of divergences make a model not well-formed

- Ontological problems can be addressed in a more abstract way, thanks to the knowledge of domain-specific concepts

- Even more interesting, ontological conflicts can be tolerated by supporting multiple development branches which will be re-unified later on after user's domain-specific decisions

Model merging



Ontological Conflict!

Current status and Outline

- We made several experiments with available platforms supporting ontology specifications
- It also exists an extension of the ATL transformation engine to consider linguistic and ontological aspects
- Both linguistic and ontological versioning can be supported by means of model-based difference representation mechanisms
- There exists a number of ontology specifications, especially in the Semantic Web research field
- The available ontologies can be extended to include OO programming, state machines, and so forth

Current status and Outline

- We are currently investigating a generic differencing engine that can be bound to domain-specific aspects

- Directly interconnected to the previous it is the creation of a conflict detection and resolution technique keeping separated linguistic and ontological aspects

- Future investigations will involve the creation of an ontology aware model coevolution mechanism

- Further investigations will be devoted to combination of several ontologies (e.g. in the case of multiview modelling systems)