

Weaving Variability into Domain Metamodels

B. Morin, G. Perrouin, Ph. Lahire,
O. Barais, G. Vanwormhoudt
and J-M. Jézéquel

MODELS'09

Variability Management session

Friday 9th October



Outline

- Introduction and Motivations
- Variability as a (meta-) Aspect
- Applications on 2 domain metamodels
- Conclusions and future works

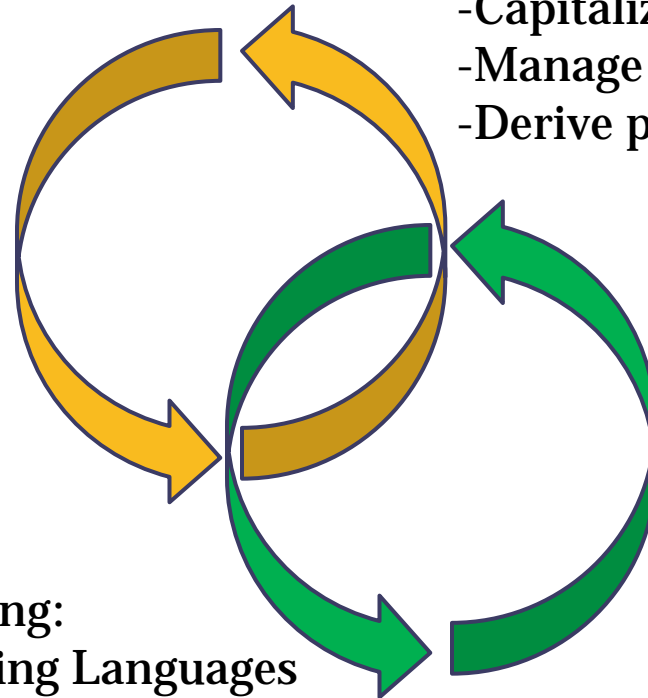
Introduction

- Products should meet (changing) user requirements
- Time-to-market is a key issue in a competitive world
- Developing products from scratch is time consuming...

SPL and MDE in concert

Software Product Lines:

- Capitalize on commonalities
- Manage variabilities
- Derive products depending on the needs



Model-Driven Engineering:

- Domain Specific Modeling Languages
- Composition, Transformation, code generation
- Early validation, simulation

MDE can support SPL
→Perrouin et al., SPLC'08

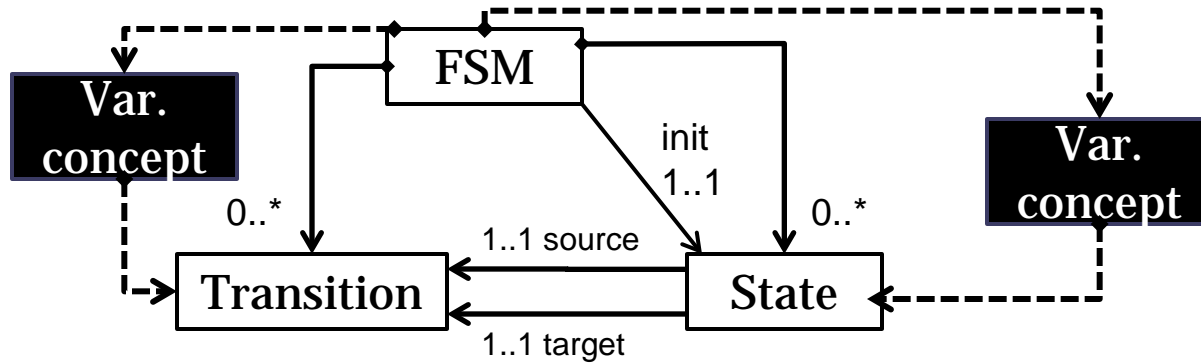
MDE can benefit from SPL
→Haugen et al., SPLC'08

Two main trends

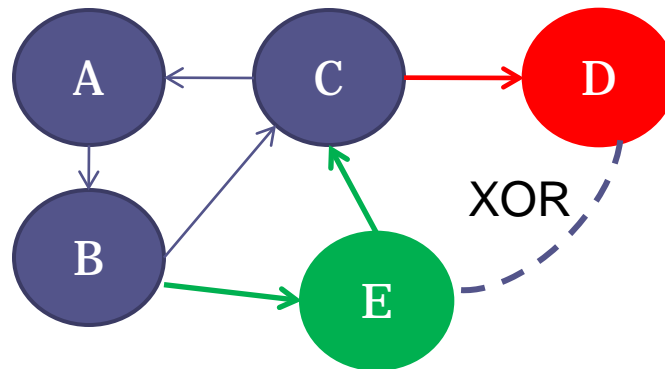
- Put variability **inside** the models
- Put variability **aside** of the models

Variability inside the models

M2

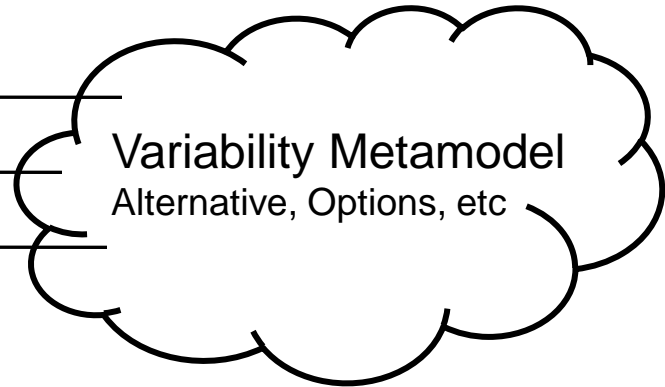
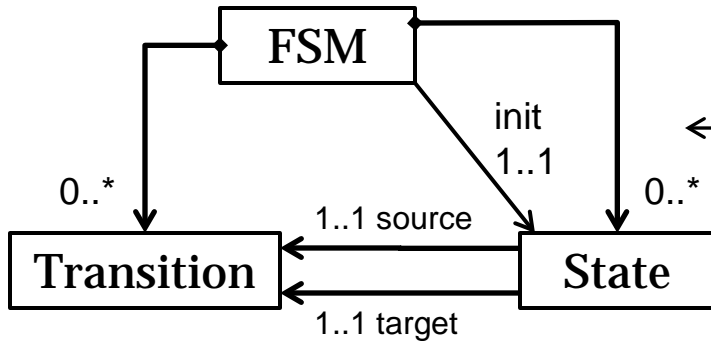


M1

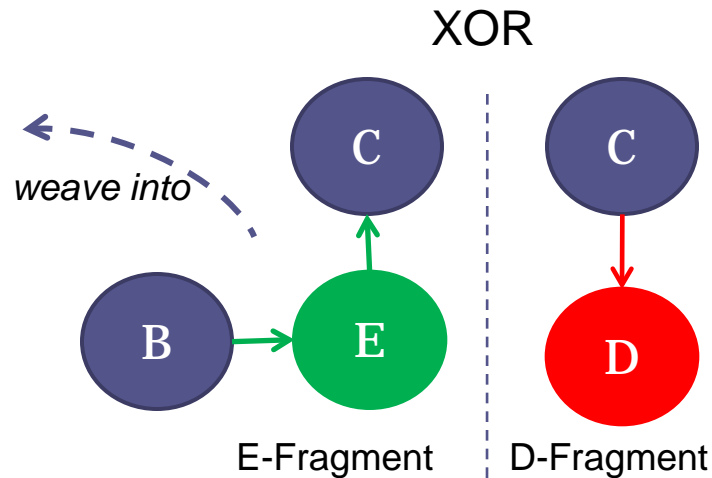
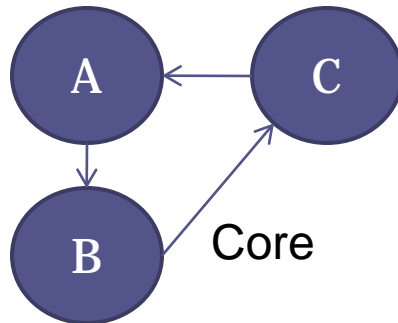


Variability aside the models

M2



M1



	++	--
Inside	<ul style="list-style-type: none"> -Variability directly rely on domain concepts 	<ul style="list-style-type: none"> -One (big) model with mandatory/variable elements -Modification of the DSML
	<ul style="list-style-type: none"> -Systematic and Flexible introduction of variability concepts, for any DSML -Limit the impact of the modification 	
Aside	<ul style="list-style-type: none"> -Clear separation of the variability -DSML remains unchanged 	<ul style="list-style-type: none"> -Extra effort for specifying composition/replacement rules, using another DSL -Co-Evolution core model / variability model → Maintenance

Outline

- Introduction and Motivations
- **Variability as a (meta-) Aspect**
- Applications on 2 domain metamodels
- Conclusions and future works

Variability

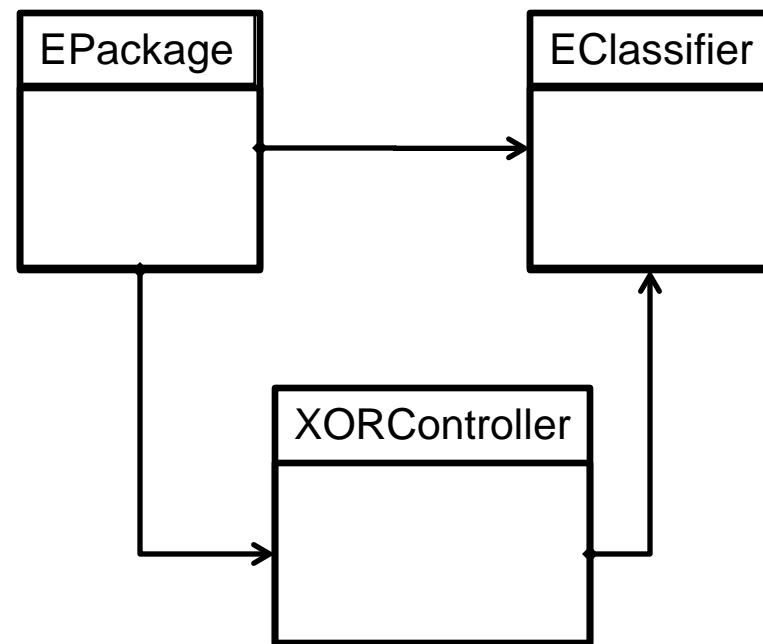
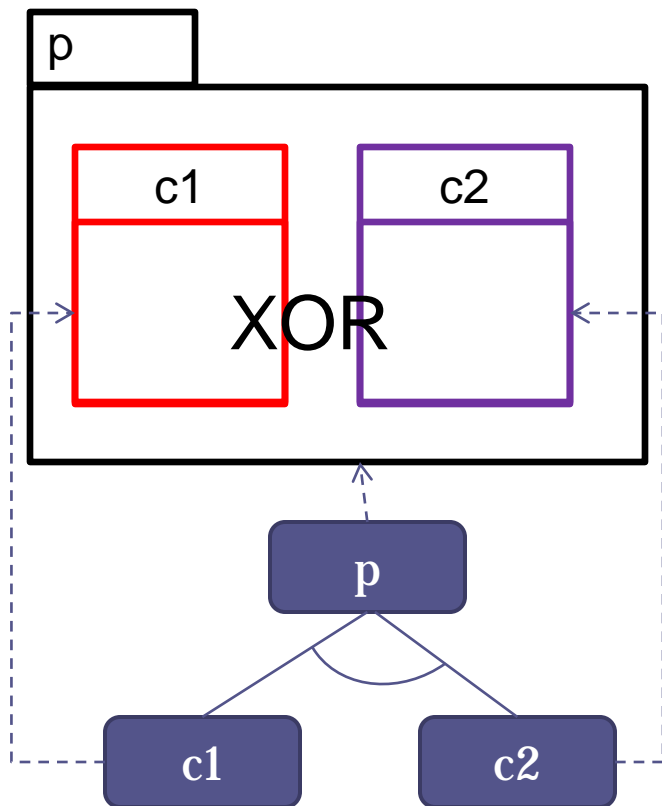
- **Orthogonal to a domain**
 - Any DSML can possibly integrate variability
- **Standard concepts**
 - Alternative (XOR), options, n-among-p choices, etc
 - Not always useful everywhere (depending on the needs/domains)
- **→ Variability is a variable concern...**

A metamodel for Variability

- Inspired by previous work (Perrouin et al., SPLC'08)
- Operators
 - Xor, Or, And, Vp (n-among-p), Optional
 - Two hierarchies of operators
 - Homogeneous (e.g, an alternative between two classes of a package)
 - Heterogeneous (composite features)
- Constraints
 - Require
 - Exclude
 - ...

Where to weave variability?

- Insert variability controllers on the references



Aspect-Oriented Metamodeling

- **Meta-Aspect: Aspect that applies to a metamodel**
- **Metamodels are models like others**
 - A (meta)model conforms to a (meta-)metamodel
- **Aspect-Oriented Metamodeling**
 - AOM applied at the metamodel level
 - Metamodels == base models for the aspects

Variability (meta-)Aspect

- Variability can be inserted in multiple places
 - Symetric merge-based AOM not well fitted
 - e.g., Kompose
 - Pointcut-based AOM more adapted
 - e.g., MATA, SmartAdapters
- Variability should not be systematically woven

SmartAdapters to weave (meta-) Aspects

- An aspect is composed of:
 - A pointcut model (**where?**)
 - An advice model (**what?**)
 - A composition protocol (**how?**)
- Before weaving, the pointcut is contextualized
 - Automatically e.g., using pattern matching
 - By hand e.g., by specifying mappings
 - Semi-automatically
 - Automatic detection
 - On-demand weaving

Variability meta-Aspect in details

- Pointcut model matches any references between 2 meta-classes.
- Controllers can be introduced on these references
 - Only useful (homogeneous) controllers are introduced
- Useful heterogeneous controllers are introduced
- Constraints are attached to the root meta-class

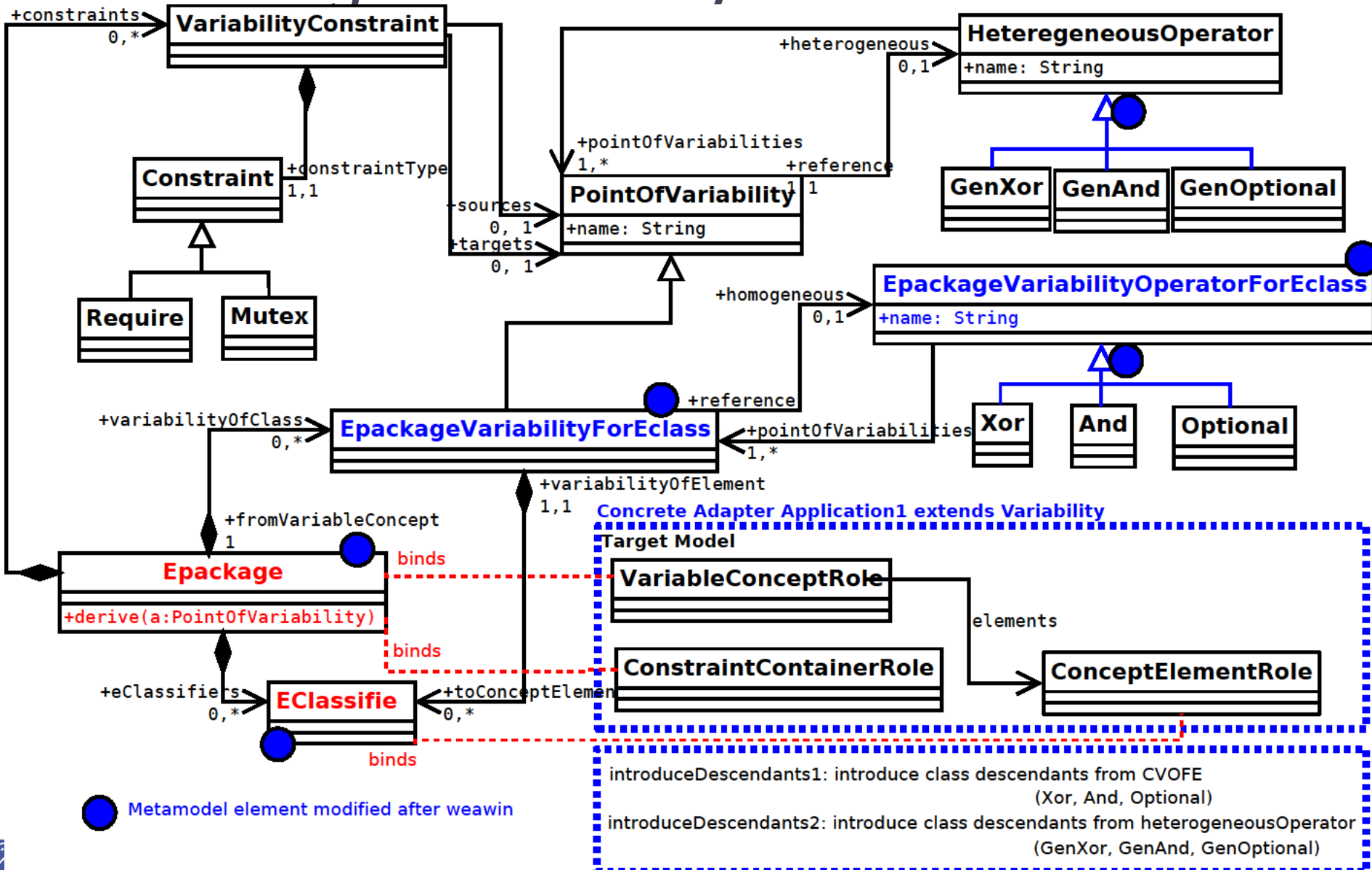
Outline

- Introduction and Motivations
- Variability as a (meta-) Aspect
- **Applications on 2 domain metamodels**
- Conclusions and future works

Class Diagram with Variability

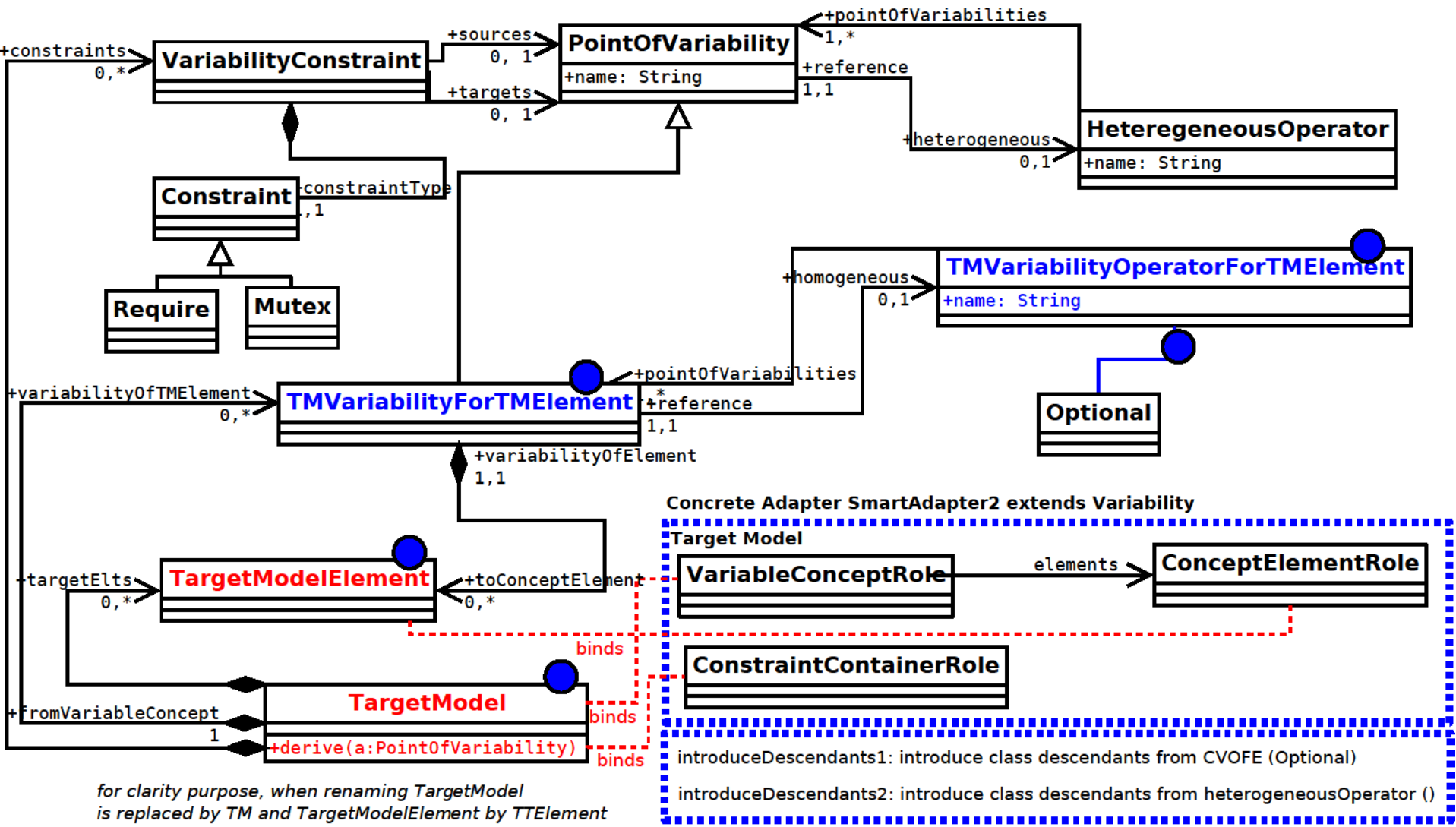
- Integrate variability into class diagrams, similar to
 - Tewfik Ziadi and Jean-Marc Jézéquel. -- *Software Product Lines*, chapter **Product Line Engineering with the UML: Deriving Products**, pages 557--586. -- Number ISBN: 978-3-540-33252-7. Springer Verlag, 2006.
 - Use of UML constructs + profiles

Weaving variability into EMF/ECore

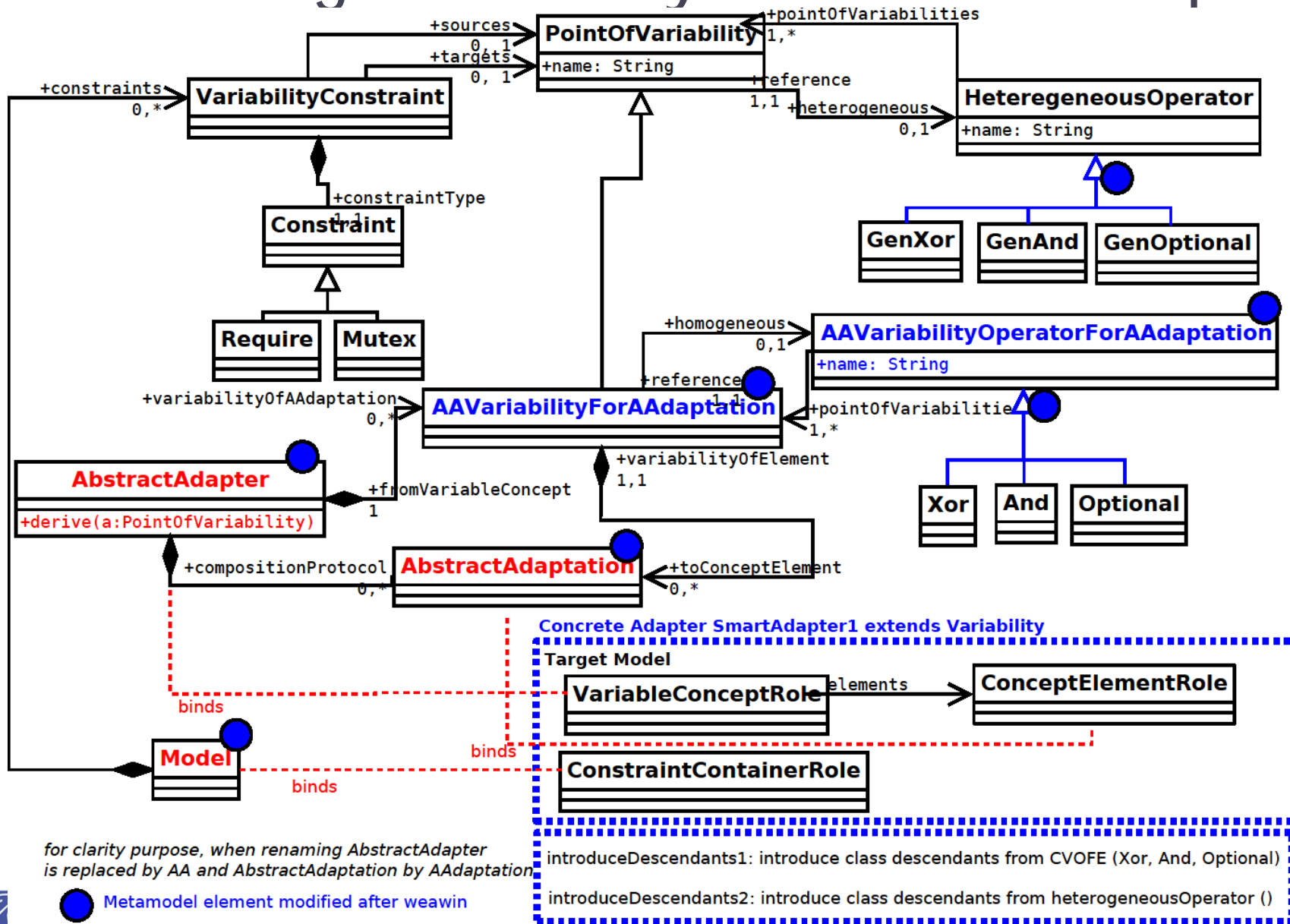


Aspect model with Variability

- Integrate variability into aspect models (weaving + matching), similar to
 - Philippe Lahire, Brice Morin, Gilles Vanwormhoudt, Alban Gaignard, Olivier Barais, and Jean-Marc Jézéquel. -- **Introducing Variability into Aspect-Oriented Modeling Approaches**. -- *In Proceedings of ACM/IEEE 10th International Conference on Model Driven Engineering Languages and Systems (MoDELS 07)*, Nashville, TN, USA, October 2007.
 - Matching variability
 - Composition variability



Weaving variability into SmartAdapters



for clarity purpose, when renaming `AbstractAdapter` is replaced by `AA` and `AbstractAdaptation` by `AAadaptation`

● Metamodel element modified after weaving

Derivation Process

- From the model (with variability)
 - Maintain a feature diagram (FD)
 - FD: Tree-like graph describing the variability
- Use existing SPL tools to select products
 - → pruned FD containing only the selected features
- From the pruned FD, derive the model
 - → a model conforming to the initial metamodel

Discussions

- A “normal” model (former MM) is a model with variability (extended MM), with no variability
 - Retro-compatibility
- A derived model (former MM) is a model with variability (extended MM), with no (more) variability
 - Still possible to use existing tools (graphical editors, checker, simulator, etc) on derived models
- Reusing graphical editors/tools to design model with variability?
 - Good editors are based on models
 - →transform these models to integrate variability

Outline

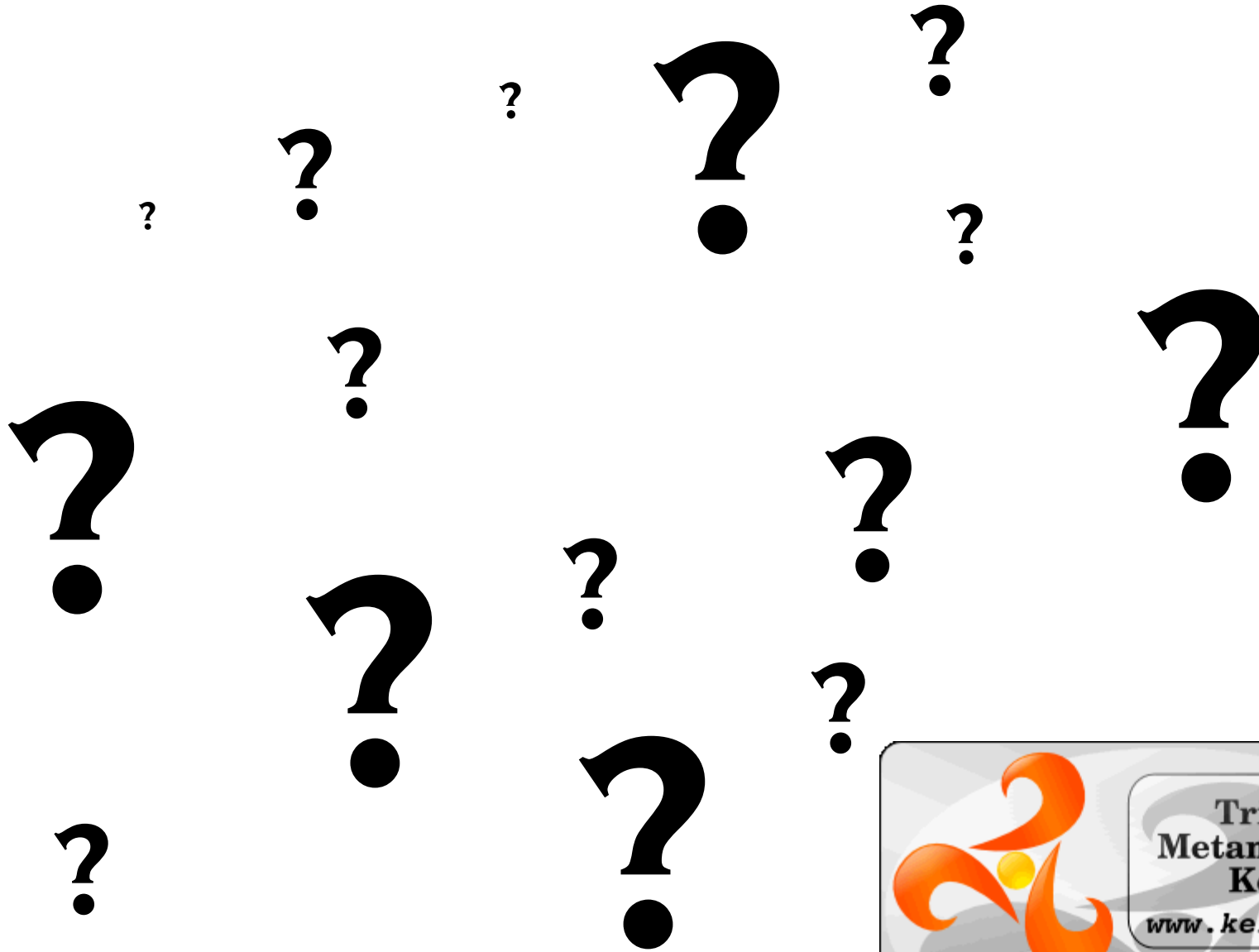
- Introduction and Motivations
- Variability as a (meta-) Aspect
- Applications on 2 domain metamodels
- **Conclusions and future works**

Conclusions

- Two main approaches to combine SPL and DSML
 - Variability Inside **Vs** Variability Aside
 - Both have ++/--
- Our approach
 - Variability Inside
 - Systematic and flexible variability weaving
 - The variability you need, where you need it

Future works

- Define other meta-aspect
 - Executability (Kermeta, MODELS'05)
 - Composability (Generic SmartAdapters)
- Link AOMM to the notion of model typing



Triskell
Metamodeling
Kernel
www.kermeta.org

Kermeta