Aspect Model Unweaving

Jacques Klein, Jörg Kienzle, Brice Morin, Jean-Marc Jézéquel

partially funded by:

the SPLIT project

fonds national de la recherche

CNRS CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

NSERC CRSNG

the DiVA project

Centre de Recherche Public Gabriel Lippmann

Département Informatique, Systèmes et Collaboration

Context: Model@runtime & AOM

➔ Critical Systems:
- Need to be continuously available
- Need to evolve at runtime

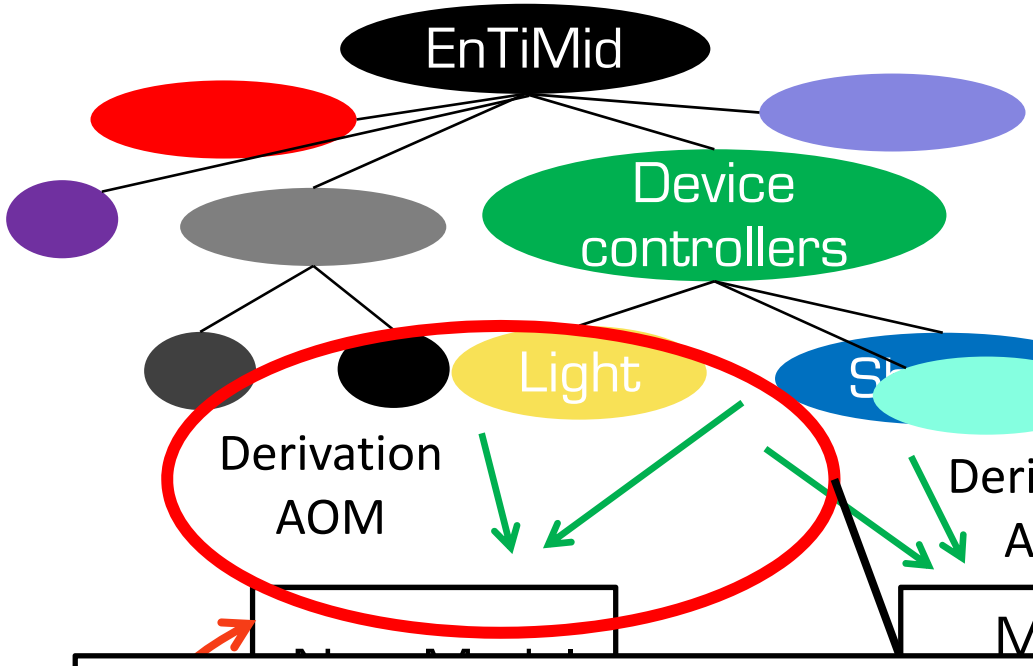
➔ Reconfiguration is a complexe and difficult task
- Need to be validated before the adaptation

A solution: use models for abstraction
(model@runtime workshops,
Morin et al. [ICSE2009])

Picture from http://www.apt.gc.ca

Running system

Context: Model@runtime, AOM, …



1) Select the most adapted architectural model

Had a flu
→ stayed in bed
Now in a better shape
→ a nurse every day

2) Vali

How improve the performance (especially time) of the Aspect-Oriented Modeling Derivation step

system

Problem and objective

➔ Problem:

Current model weavers do not support unweaving: Removing one aspect from a configuration with n aspects requires re-weaving of n-1 aspects.

➔ Objective of this talk:

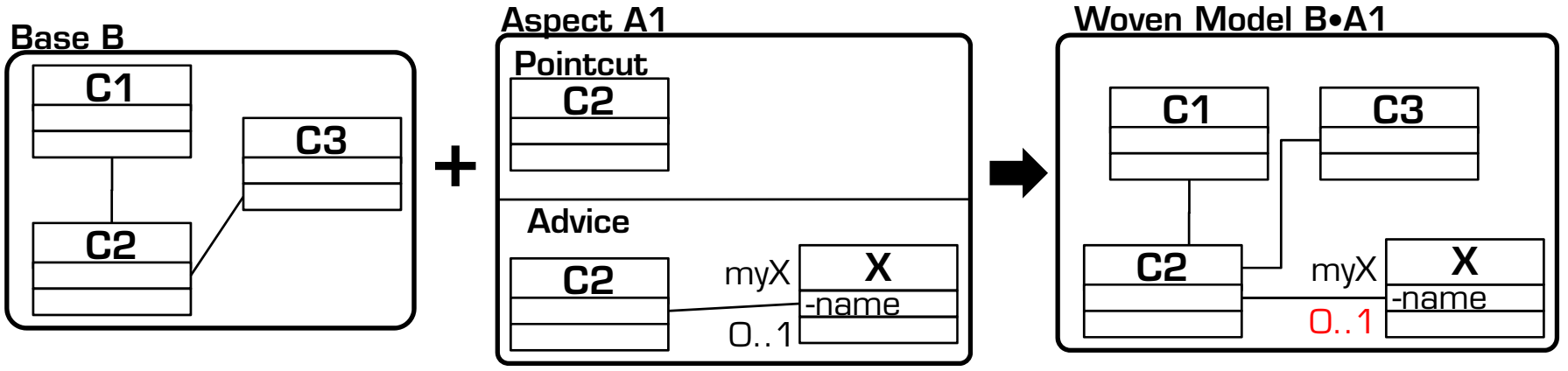
Show how an aspect can be unwoven from a model previously generated by weaving, by applying a sequence of simple model modification operations

Agenda

- **Background**
- Definitions: weaving and unweaving
- Unweaving through examples…
  - Independent Aspects
  - Additive Aspects
  - Intrusive Aspects
- Using traceability for aspect classification and aspect unweaving
- Conclusion and future work

Background: GeKo

- GeKo is a Generic Aspect Model Weaver which can be used for any modeling formalism with a well-defined meta-model
- The weaving process is two-phased.
  - The first step consists in the detection of the join points corresponding to the Pointcut diagram.
  - The second step consists in the composition of the advice model with the base model at the level of the join points previously detected



Operation-Based Model Construction

➔ Instead of defining a model as a set of objects or model elements, we use the approach presented by Blanc et al. [ICSE'08] where a model is defined by a sequence of elementary construction operations.

1) ***create(me,mc)*** corresponds to the creation of a model element instance *me* of the meta-class *mc*;
2) ***delete(me)*** corresponds to the deletion of the model element instance *me*;
3) ***setProperty(me,p,Values)*** corresponds to the assignment of a set of *Values* to the property *p* of the model element *me*;
4) ***setReference(me,r,References)*** corresponds to the assignment of *References* to the reference *r* of the model element *me*.

Agenda

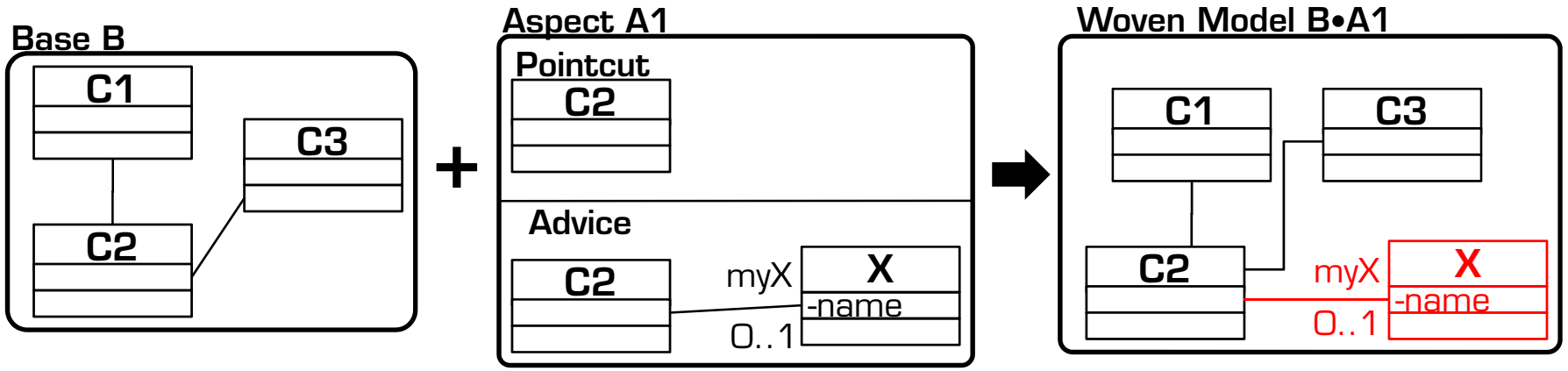
- ➔ Background
- ➔ **Definitions: weaving and unweaving**
- ➔ Unweaving through examples…
  - Independent Aspects
  - Additive Aspects
  - Intrusive Aspects
- ➔ Using traceability for aspect classification and aspect unweaving
- ➔ Conclusion and future work

Definitions: Weaving

➔ Weaving of A_i at the level of a join point jp

$$weave(A_i, \text{jp}) = \sigma^1_{\text{jp},1} \bullet \sigma^1_{\text{jp},2} \bullet \ldots \bullet \sigma^1_{\text{jp},k}$$

➔ Example:



$$weave(A_1, mp) = create(X, EClass) \bullet setProperty(X, name, \{X\}) \bullet$$
$$create(nameAtt, EAttribute) \bullet setProperty(nameAtt, name, \{\text{“}name''\}) \bullet$$
$$setReference(X, EAttribute, \{nameAtt\}) \bullet create(ref, EReference) \bullet \ldots$$

Definitions: Weaving

➔ Weaving of A_i at the level of a join point jp

$$weave(A_i, \text{jp}) = \sigma^i_{\text{jp},1} \bullet \sigma^i_{\text{jp},2} \bullet \ldots \bullet \sigma^i_{\text{jp,k}}$$

➔ Weaving of A_i for all the join points:

$$\begin{aligned} weave(A_i) &= weave(A_i, mp_1) \bullet weave(A_i, mp_2) \bullet \ldots \bullet weave(A_i, mp_h) \\ &= \sigma^i_{mp_1,1} \bullet \ldots \bullet \sigma^i_{mp_1,k} \bullet \sigma^i_{mp_2,1} \bullet \ldots \bullet \sigma^i_{mp_2,k} \bullet \ldots \bullet \sigma^i_{mp_h,1} \bullet \ldots \bullet \sigma^i_{mp_h,k} \end{aligned}$$

➔ Weaving of a sequence of aspects A_1, A_2, …, A_n

$$weave(A_1, A_2, \ldots, A_n) = weave(A_1) \bullet weave(A_2) \bullet \ldots \bullet weave(A_n)$$

Definitions: Unweaving

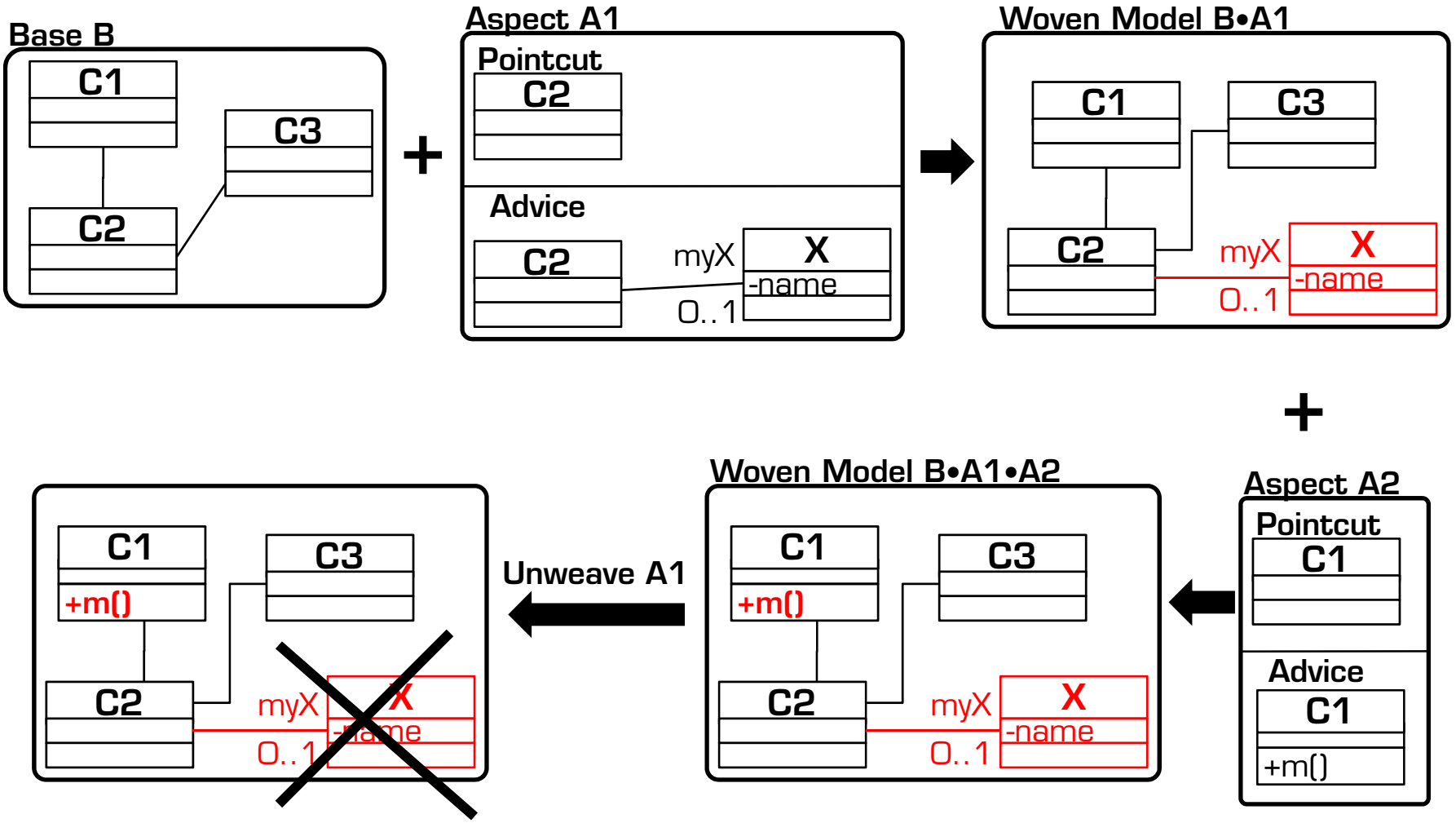
➔ For a sequence of aspects A_1, A_2, …, A_n, the unweaving of an aspect A_i corresponds to the model obtained by the weaving of the initial sequence but omitting A_i.

$$unweaving(A_i) = \begin{cases} weave(A_2, A_3, ..., A_n) & i = 1 \\ weave(A_1, A_2, ..., A_{i-1}, A_{i+1}, ..., A_n) & 1 < i < n \\ weave(A_1, A_2, ..., A_{n-1}) & i = n \end{cases}$$

Agenda

- ➔ Background
- ➔ Definitions: weaving and unweaving
- ➔ **Unweaving through examples…**
  - **Independent Aspects**
  - **Additive Aspects**
  - **Intrusive Aspects**
- ➔ Using traceability for aspect classification and aspect unweaving
- ➔ Conclusion and future work

A simple example of Unweaving: Independent Aspects



What are independent aspects?

➔ For a sequence of aspects A_1, A_2, …, A_n, an aspect A_i is independent of the aspects $A_{k,k>i}$ that are woven after A_i when A_i :

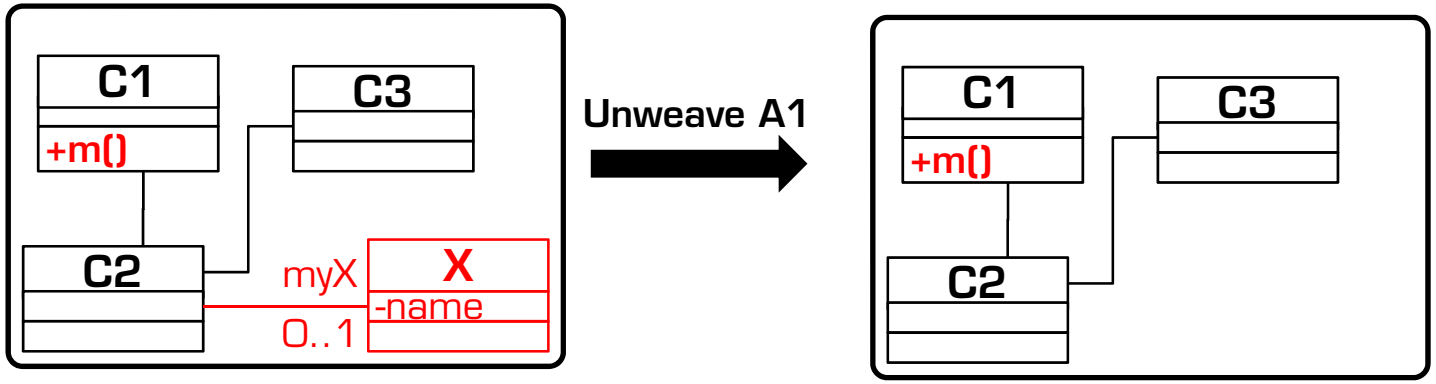
- Neither introduces model elements which were used in a join point of one of the aspects $A_{k,k>i}$
- Nor removes model elements which could have formed a join point for one of the aspects $A_{k,k>i}$
- Nor changes any properties or references that were used or could have been used in a join point of one of the aspects $A_{k,k>i}$

How to unweave independent aspects?

➔ Unweaving of aspect A_i simply consists in ***undoing*** the weave operation, i.e., in applying, for each join point, the inverse construction operations in the opposite order of the sequence defined by weave*(A_i, jp)*.

$\sigma_{mp,j}$	**inverse($\sigma_{mp,j}$)**
create(me,mc)	delete(me)
delete(me)	create(me,mc)
setProperty(me,p,value)	If ∃*setProperty(me,p,value')*∈ weave($A_{k,k<i}$) then *setProperty(me,p,value')* else *setProperty(me,p,∅)*
setRef(me,p,value)	If ∃*setRef(me,p,value')*∈ weave($A_{k,k<i}$) then *setRef(me,p,value')* else *setRef(me,p,∅)*

Example of unweaving: Independent Aspects

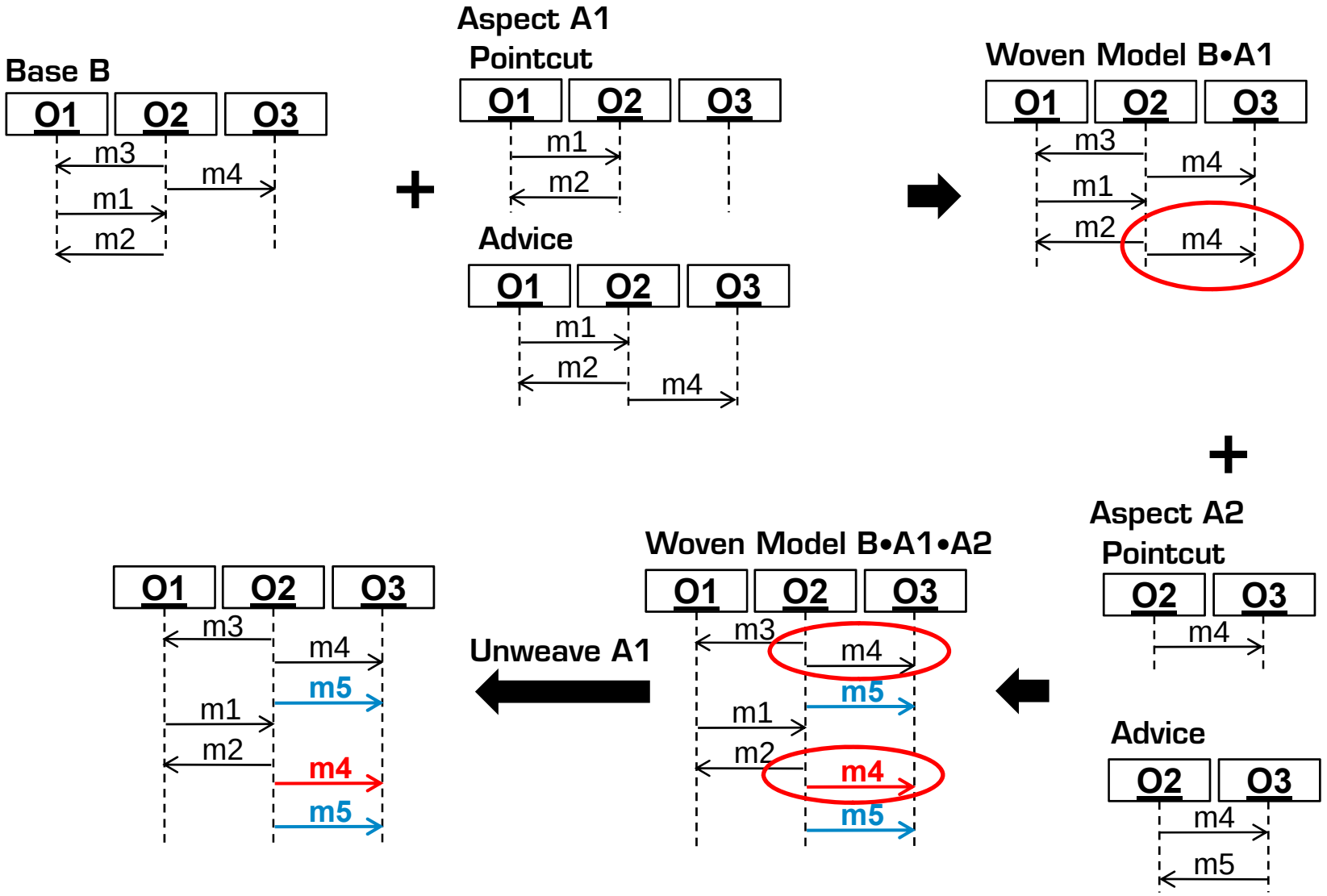


Remove the reference

$$unweave(A_1) = undo(A_1) = setReference(C2, EReference, \emptyset) \bullet setProperty(ref, EType, \emptyset) \bullet setProperty(ref, name, \emptyset) \bullet delete(ref) \bullet setReference(X, EAttribute, \emptyset) \bullet setProperty(nameAtt, name, \emptyset) \bullet delete(nameAtt, \emptyset) \bullet setProperty(X, name, \emptyset) \bullet delete(X)$$

Remove the class X

Example of Additive Aspects



What are additive aspects?

➔ For a sequence of aspects A_1, A_2, …, A_n, an aspect A_i is additive for the aspects $A_{k,k>i}$ that are woven after A_i when A_i :

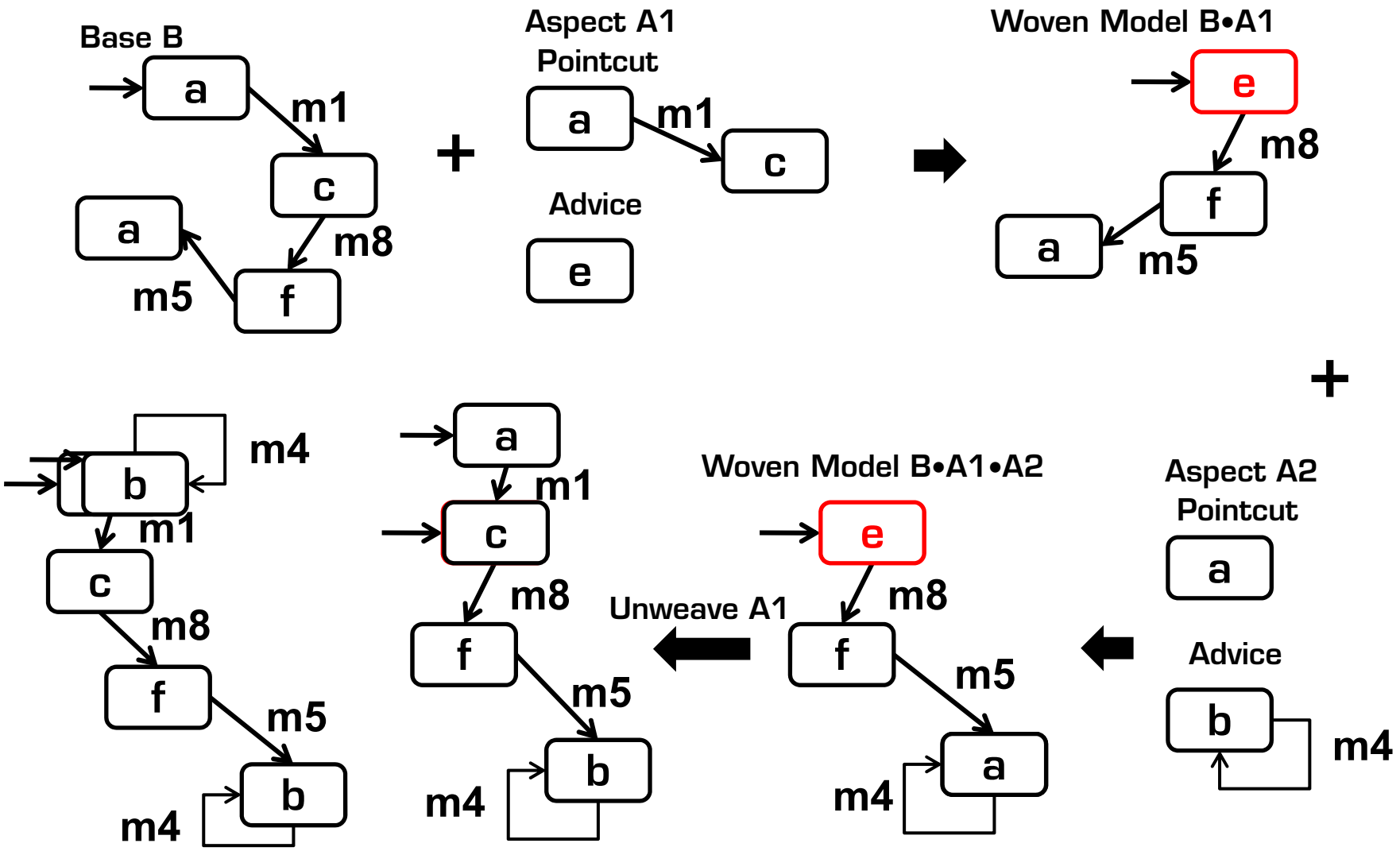
- Introduces model elements which were used in a join point of one of the aspects $A_{k,k>i}$
- But doesn't remove model elements which could have formed a join point for one of the aspects $A_{k,k>i}$

How to unweave additive aspects?

➔ For a sequence of aspects A_1, A_2, …, A_n, and an aspect A_i which is additive for the aspects $A_{k,k>i}$

- the unweaving of the aspect does not simply consist in undoing the weave operation of A_i
- the unweaving operation has to also undo the weaving of all advice of aspects $A_{k,k>i}$ that were woven because of a join point that contained elements that *Ai* added

Example of Intrusive Aspects (General Aspect)



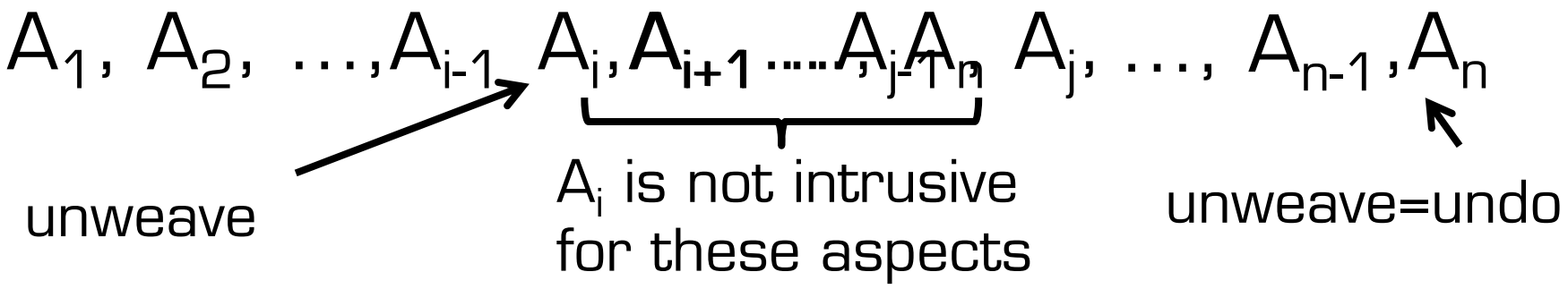
What are intrusive aspects?

➔ For a sequence of aspects A_1, A_2, …, A_n, an aspect A_i is intrusive for the aspects $A_{k,k>i}$ that are woven after A_i when A_i :

- As soon as A_i removes model elements that could have been used to form a join point of an aspect $A_{k,k>i}$.

How to unweave intrusive aspects?

- In this case, we have to launch the weaving process again to be sure to not miss some join points.
- But to be more efficient, the idea is (to unweave an aspect A_i):



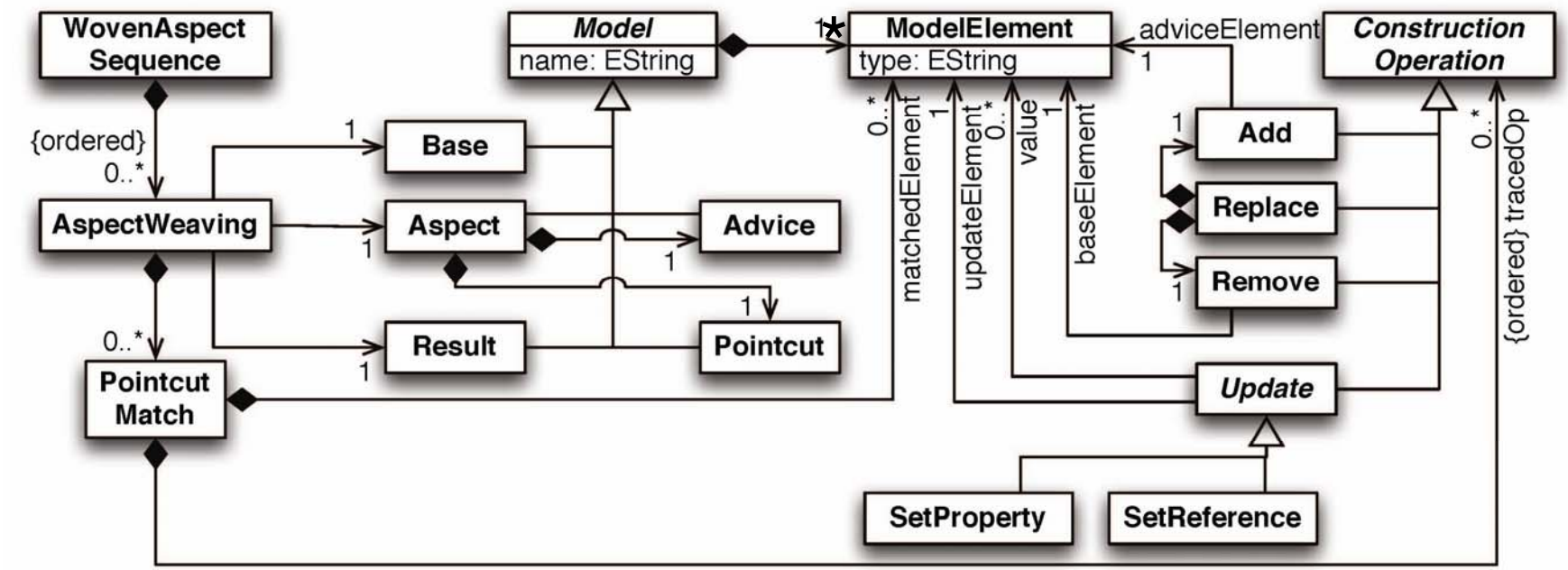
1) Compute the index j>i
2) Unweave A_n then A_{n-1}… until A_j (for these aspects, unweave=undo)
3) Unweave A_i (by applying the previous algorithms)
4) Weave again the aspects $A_{k,k>=j}$.

Agenda

- ➔ Background
- ➔ Definitions: weaving and unweaving
- ➔ Unweaving through examples…
  - Independent Aspects
  - Additive Aspects
  - Intrusive Aspects
- ➔ **Using traceability for aspect classification and aspect unweaving**
- ➔ Conclusion and future work

How to classify aspects?

➔ We use a traceability model based on a aspect traceability metamodel for GeKo



➔ Traceability model is used to record the weaving operations for all aspects and all join points

How to classify aspects?

- ➔ The traceability model is used to determine the impact of an aspect A_i on the following aspects
- ➔ Allow to classify an aspect:
  - For instance, the traceability model allows to determine conditions such as:

If $\exists$delete(elt)$\in$weave(A_i,jp) such as *elt* corresponds to a model element of a pointcut of $A_{k,k>i}$, then A_i is intrusive

Overview of the Classification Algorithm

Input: aspect A_i, join point *jp*, the traceability model corresponding to the sequence of aspects $A_1, A_2, \ldots, A_n$ in a base model *B*

Output: classification of A_i

- **If** A_i removes model elements which could have formed a join point for one of the aspects $A_{k,k>i}$ **then**

  A_i is *intrusive*

- **Else**
  - **If** A_i introduces model elements which were used in a join point of one of the aspects $A_{k,k>l}$ **then**
    - A_i is *additive*
  - **Else** A_i is *independent*

Agenda

- ➔ Background
- ➔ Definitions: weaving and unweaving
- ➔ Unweaving through examples…
  - Independent Aspects
  - Additive Aspects
  - Intrusive Aspects
- ➔ Using traceability for aspect classification and aspect unweaving
- ➔ **Conclusion and future work**

Conclusion

➔ Method to efficiently unweave an aspect A_i from a sequence of aspects A_1, A_2, …, A_n woven into a base model *B*.

- Based on the use of a traceability model recording construction operations at weave-time.
- The traceability model is used to determine the relation between A_i and the aspects which follow.
- Depending on this relation, the unweaving is more or less complicated
- In the best cases, the unweaving can be performed by directly applying a set of *undo* operations

Remark and future work

- ➔ Although the method described is presented in the context of the GeKo weaver, this method can be easily used with other weavers or tools, once an appropriate traceability model is defined

- ➔ Plans are to apply and evaluate the performance of our unweaving method in the context of adaptive systems

Question?