



# Open Aspects

Robert Hirschfeld

DoCoMo Euro-Labs

Stefan Hanenberg

University of Duisburg-Essen

ESUG, Brussels, 2005-08-16

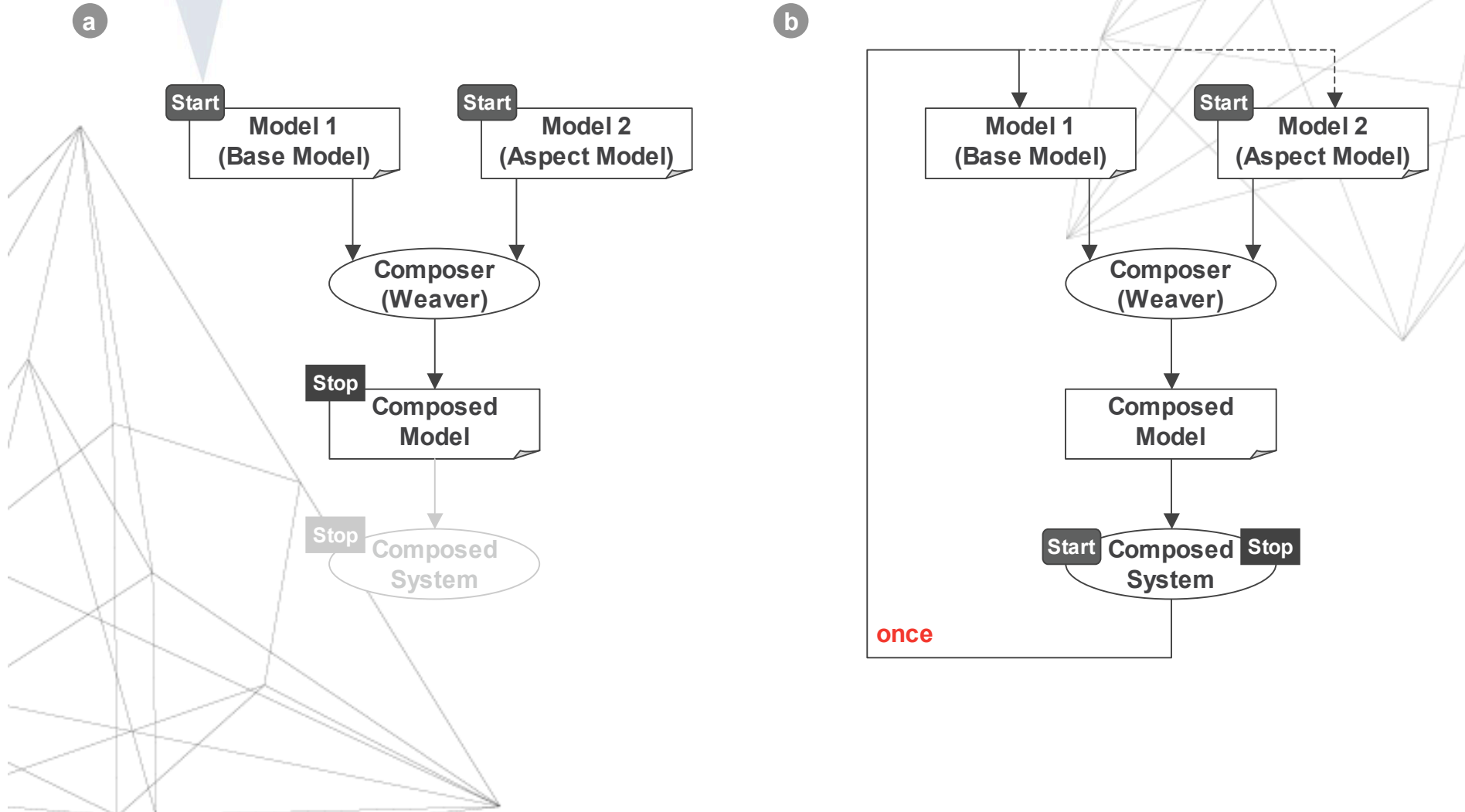


Fig. 1

MorphicMousingAspect>>adviceMouseEnter

↑ BeforeAfterAdvice

qualifier: (AdviceQualifier

attributes: { #receiverClassSpecific. })

pointcut: [

Morph withAllSubclasses

select: [:m | m includesSelector: #mouseEnter:]

thenCollect: [:m | AsJoinPointDescriptor

targetClass: m targetSelector: #mouseEnter:]

beforeBlock: [:receiver :arguments :aspect :client |

Transcript show: '\*Enter\*', arguments first printString]

“aspect lifecycle in a nutshell”

| anAspect |

anAspect ← MorphicMousingAspect new.

anAspect *install*.

anAspect *uninstall*.

Fig. 3



### MorphicMousingAspect

Morph subclass: #**MouseEnterLeaveMorph**

...

MouseEnterLeaveMorph>>mouseEnter: evt  
self beepPrimitive.

MouseEnterLeaveMorph>>mouseLeave: evt  
self beep.

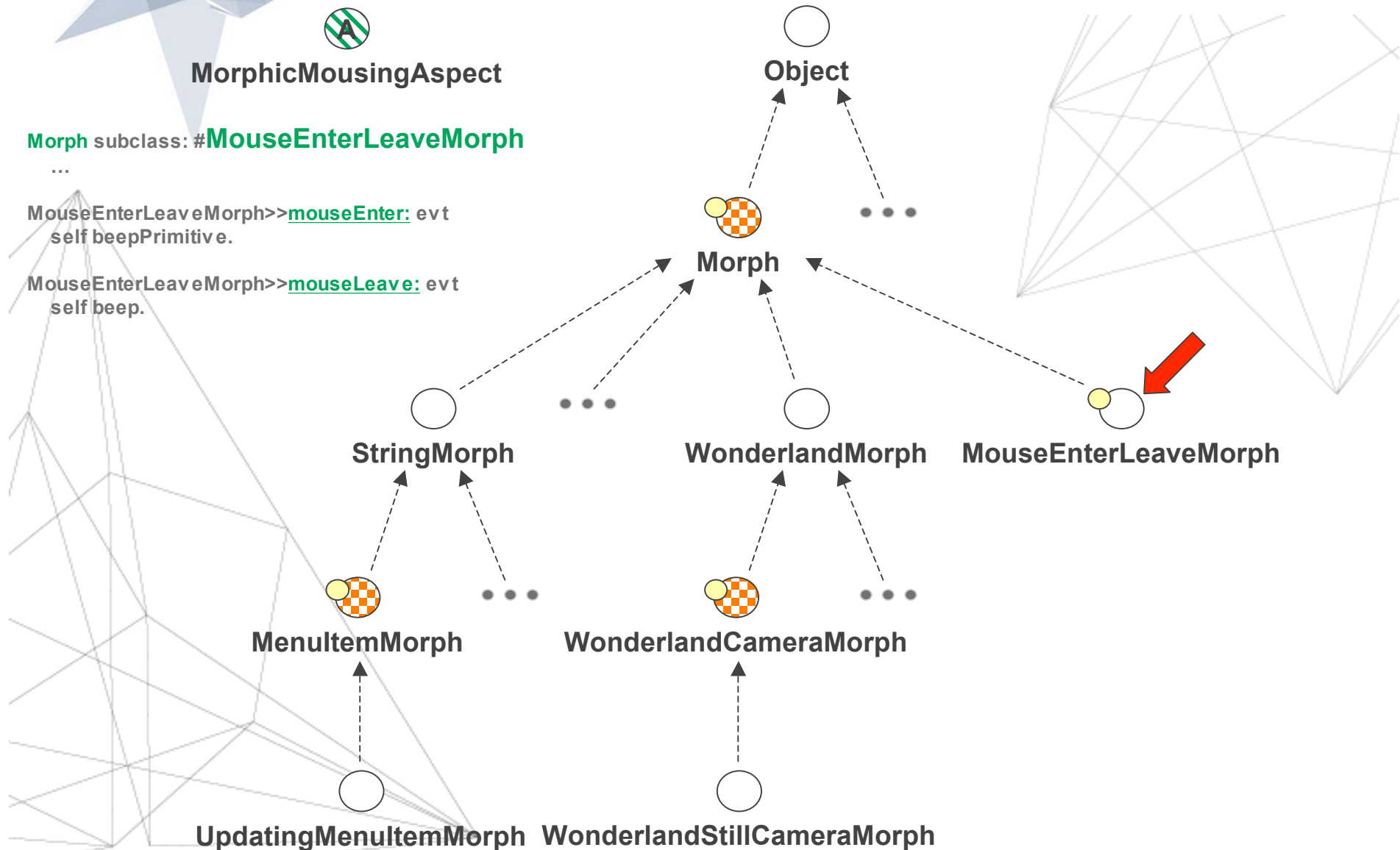


Fig. 6

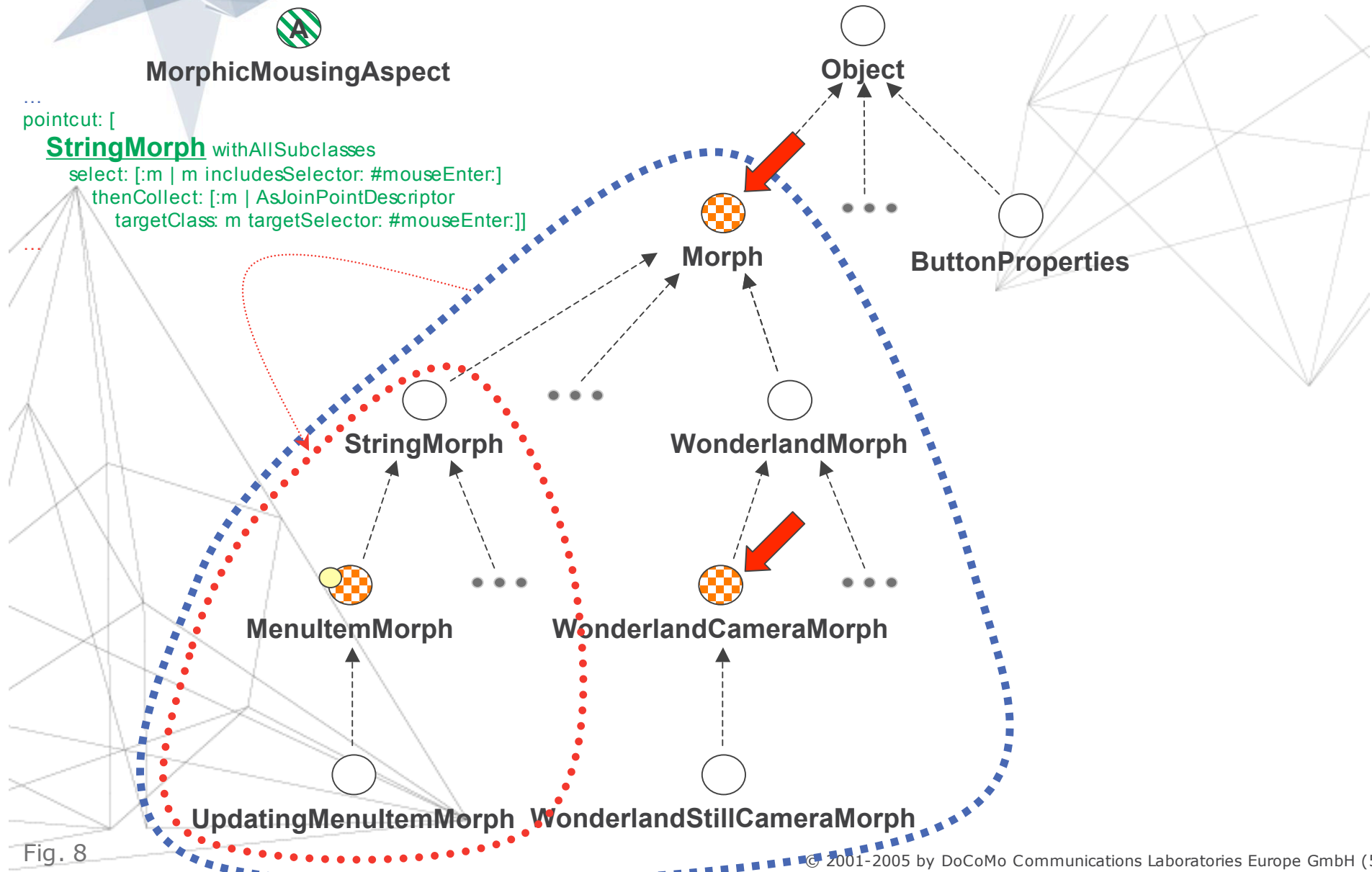


Fig. 8



MorphicMousingAspect

```
...  
beforeBlock [:receiver :arguments :aspect :client |  
Logger count increment.  
Transcript show: '*Enter*', arguments first printString]  
...
```

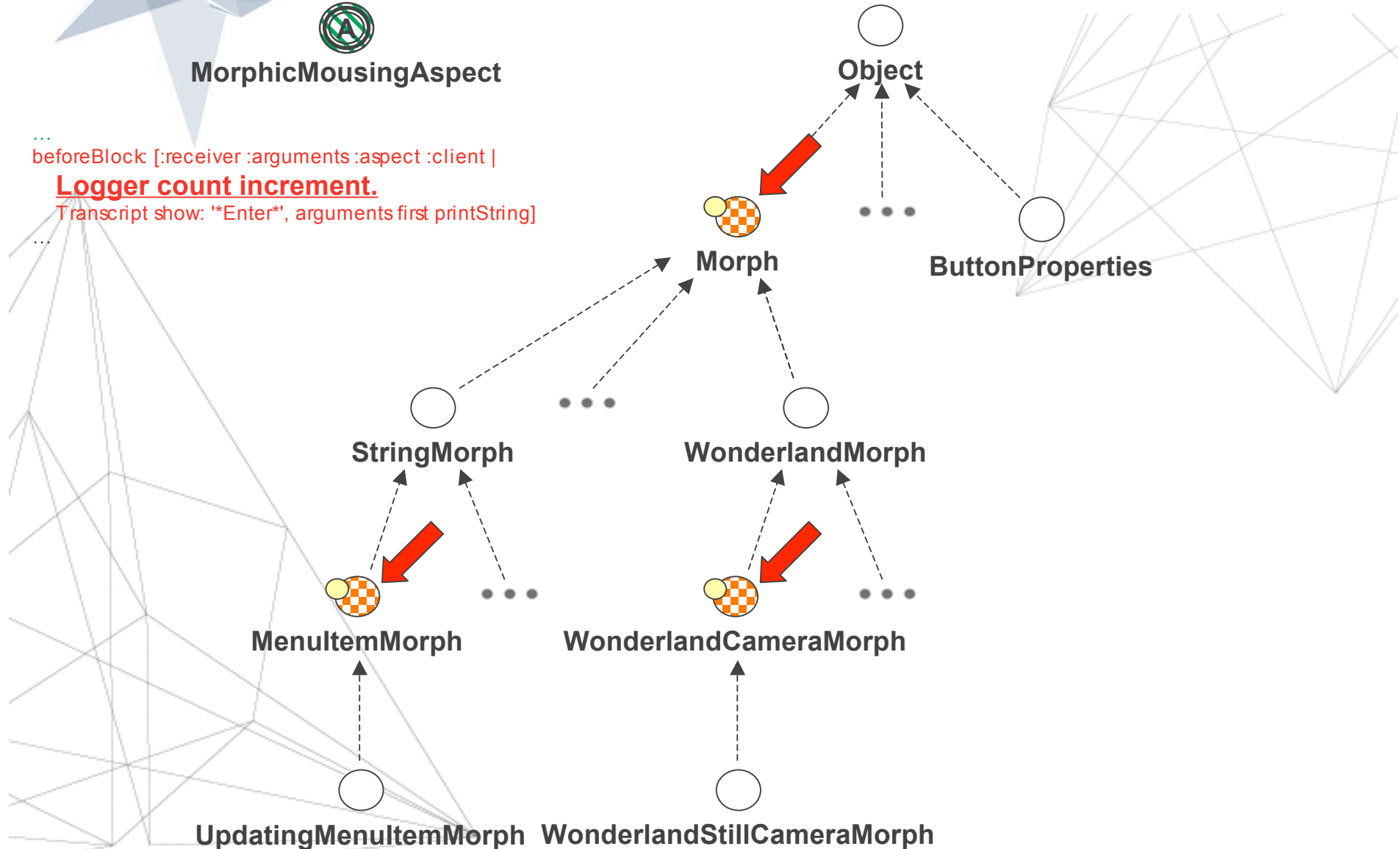


Fig. 10

# Mobile Adventure Open Aspects

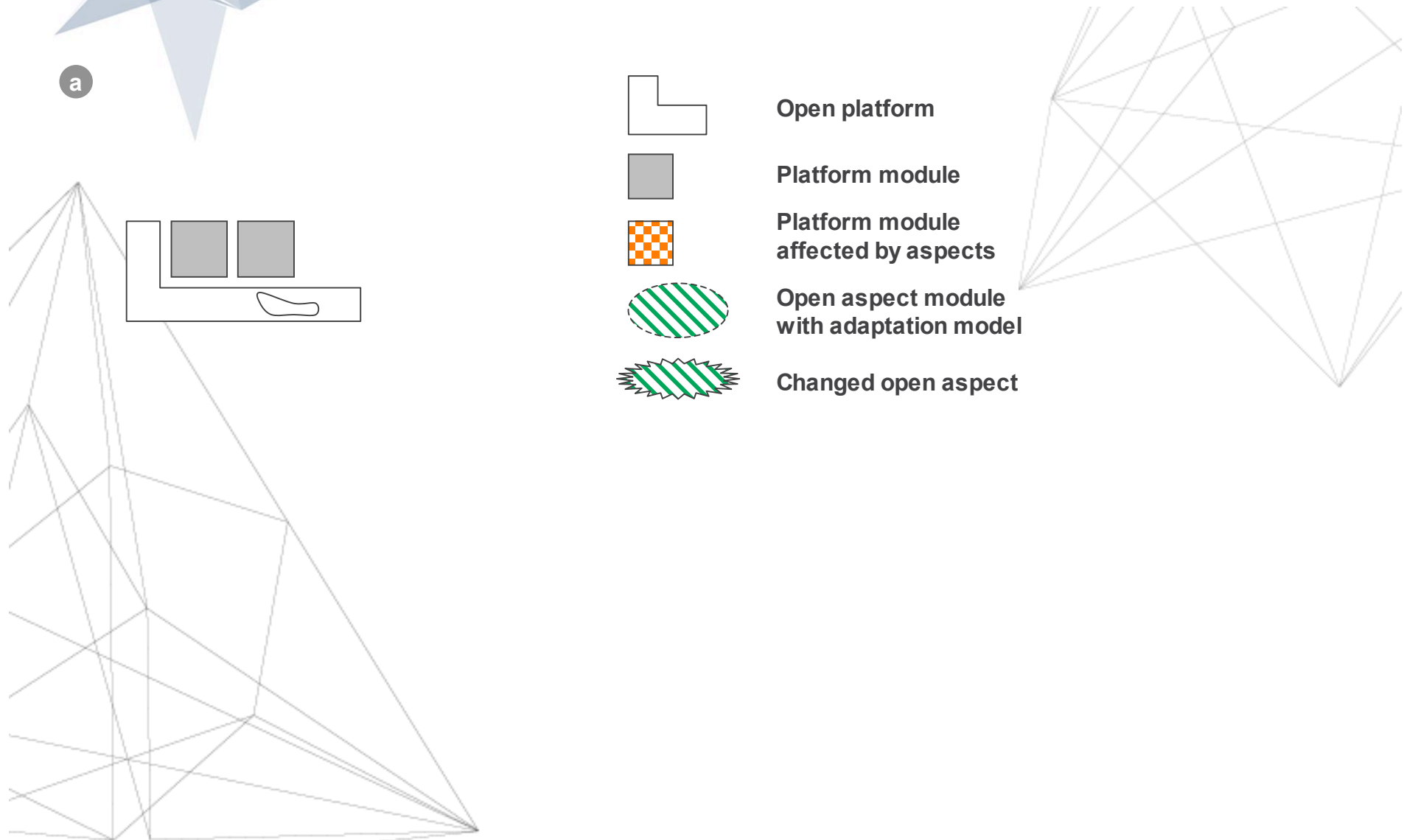


Fig. 12

# Mobile Adventure

## Open Aspects (Additive Change)

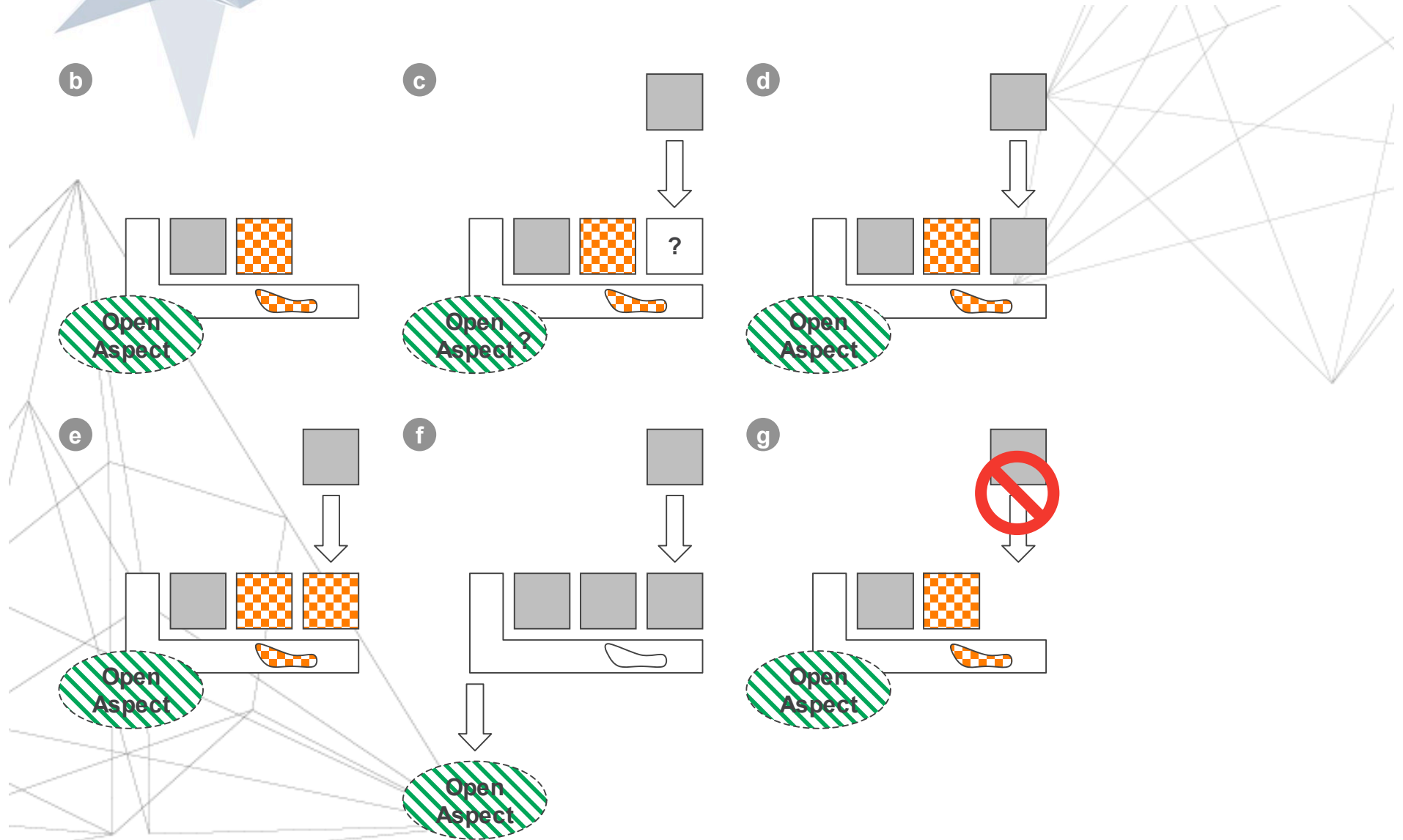


Fig. 13



# Mobile Adventure

## Open Aspects (Subtractive Change)

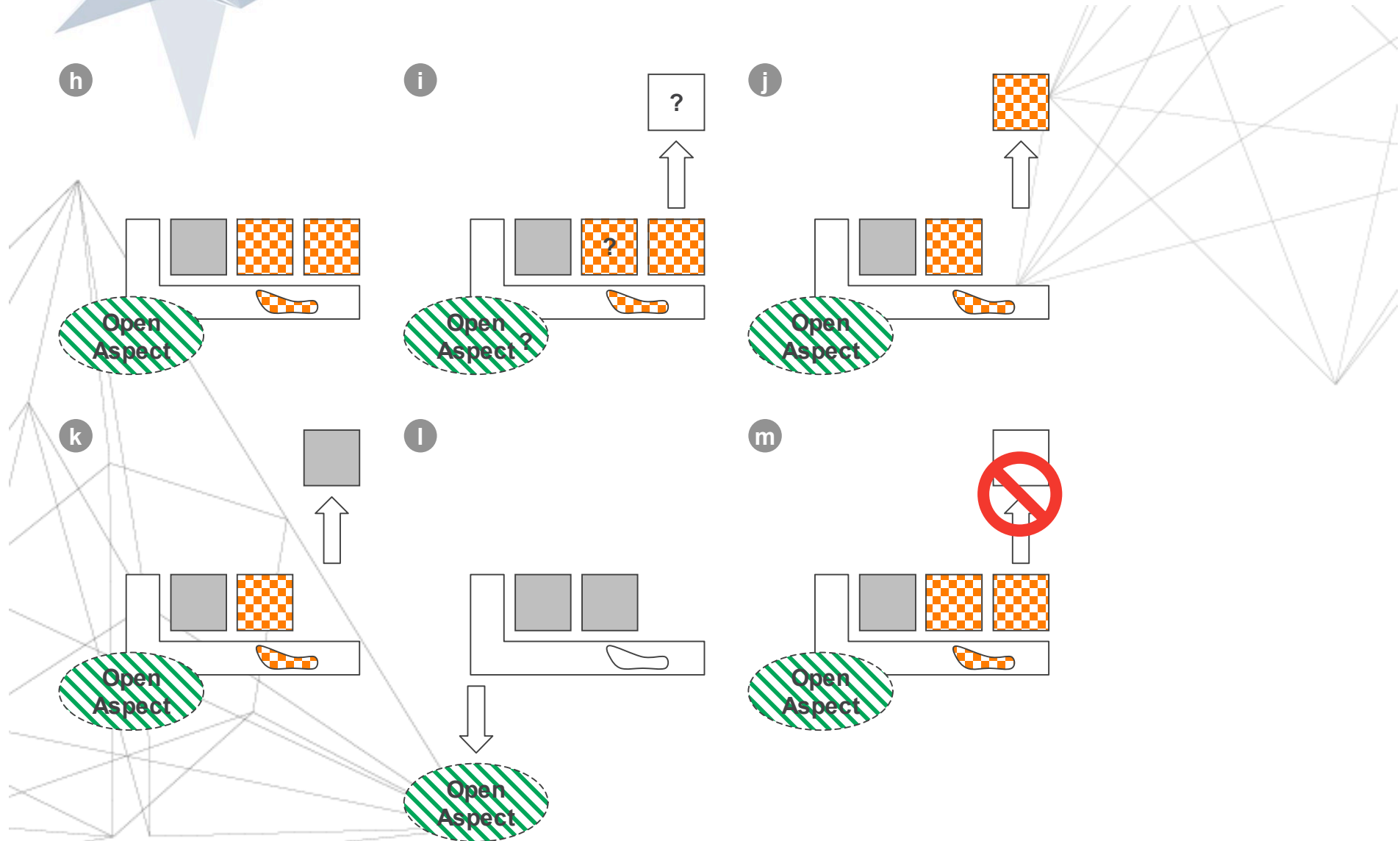


Fig. 14

# Mobile Adventure

## Open Aspects (Pointcut/Advice Change)

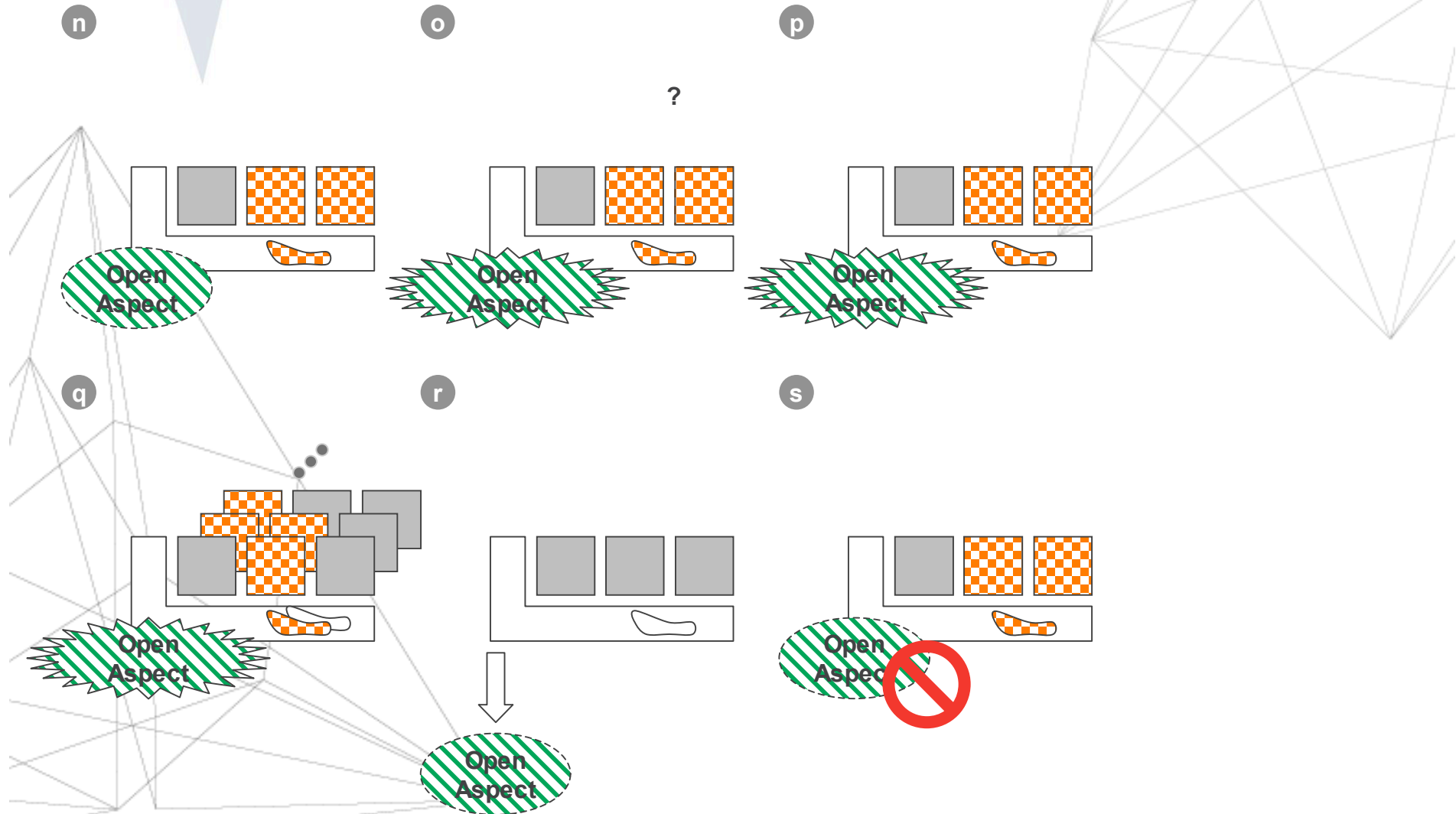


Fig. 15

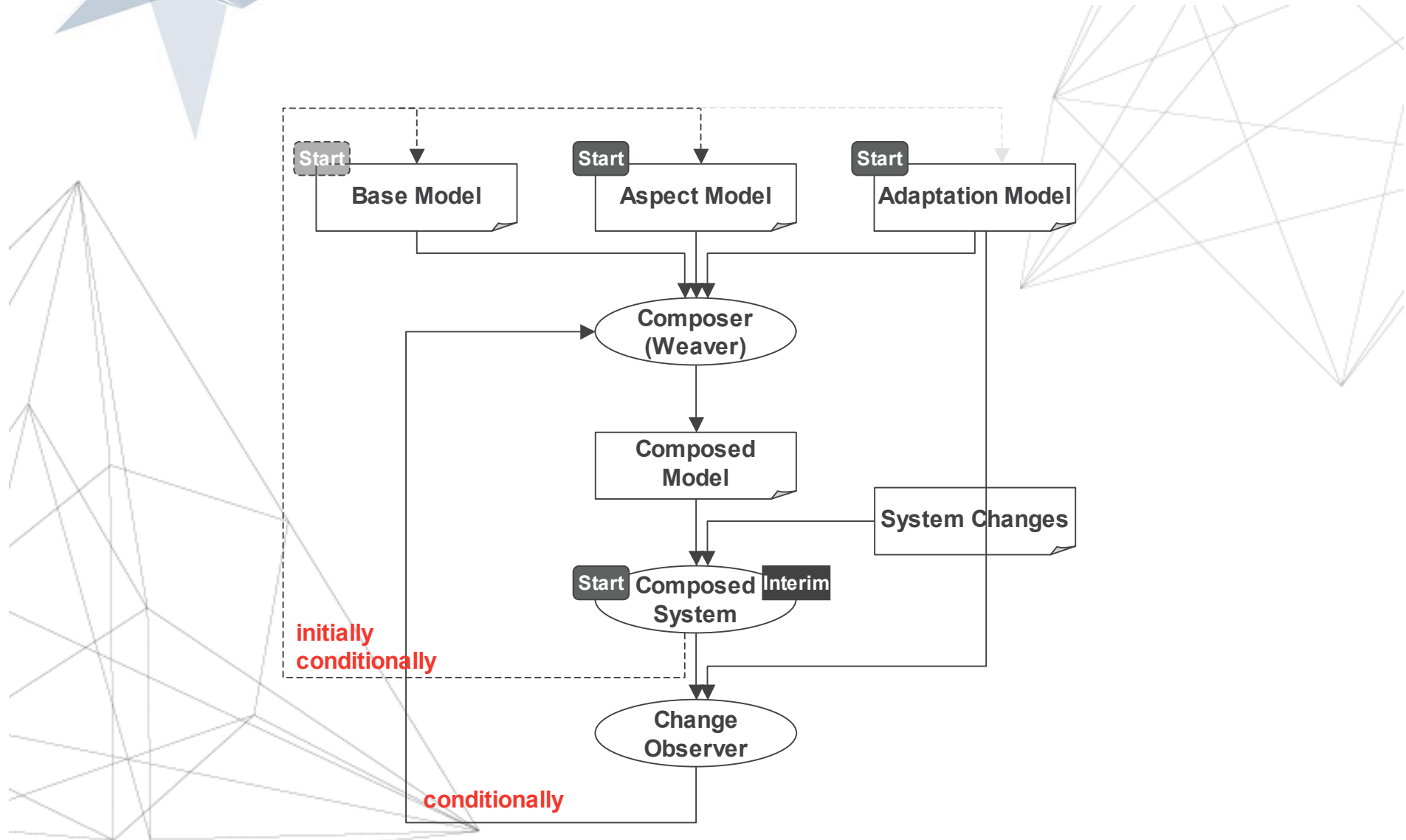


Fig. 11

# Mobile Adventure

## Advice in OpenAspectS

MorphicMousingOpenAspect>>adviceMouseEnter

↑ BeforeAfterAdvice

qualifier: (AdviceQualifier

attributes: { #receiverClassSpecific. }

adaptations: { #reinstallAdvice. }

pointcut: [

Morph withAllSubclasses

select: [:m | m includesSelector: #mouseEnter:]

thenCollect: [:m | AsJoinPointDescriptor

targetClass: m targetSelector: #mouseEnter:]

beforeBlock: [:receiver :arguments :aspect :client |

Transcript show: '\*Enter\*', arguments first printString]

Fig. 19

# Mobile Adventure With Open Aspects (Additive Change)



MorphicMousingOpenAspect

Morph subclass: #MouseEnterLeaveMorph

MouseEnterLeaveMorph>>mouseEnter: evt  
self beepPrimitive.

MouseEnterLeaveMorph>>mouseLeave: evt  
self beep.

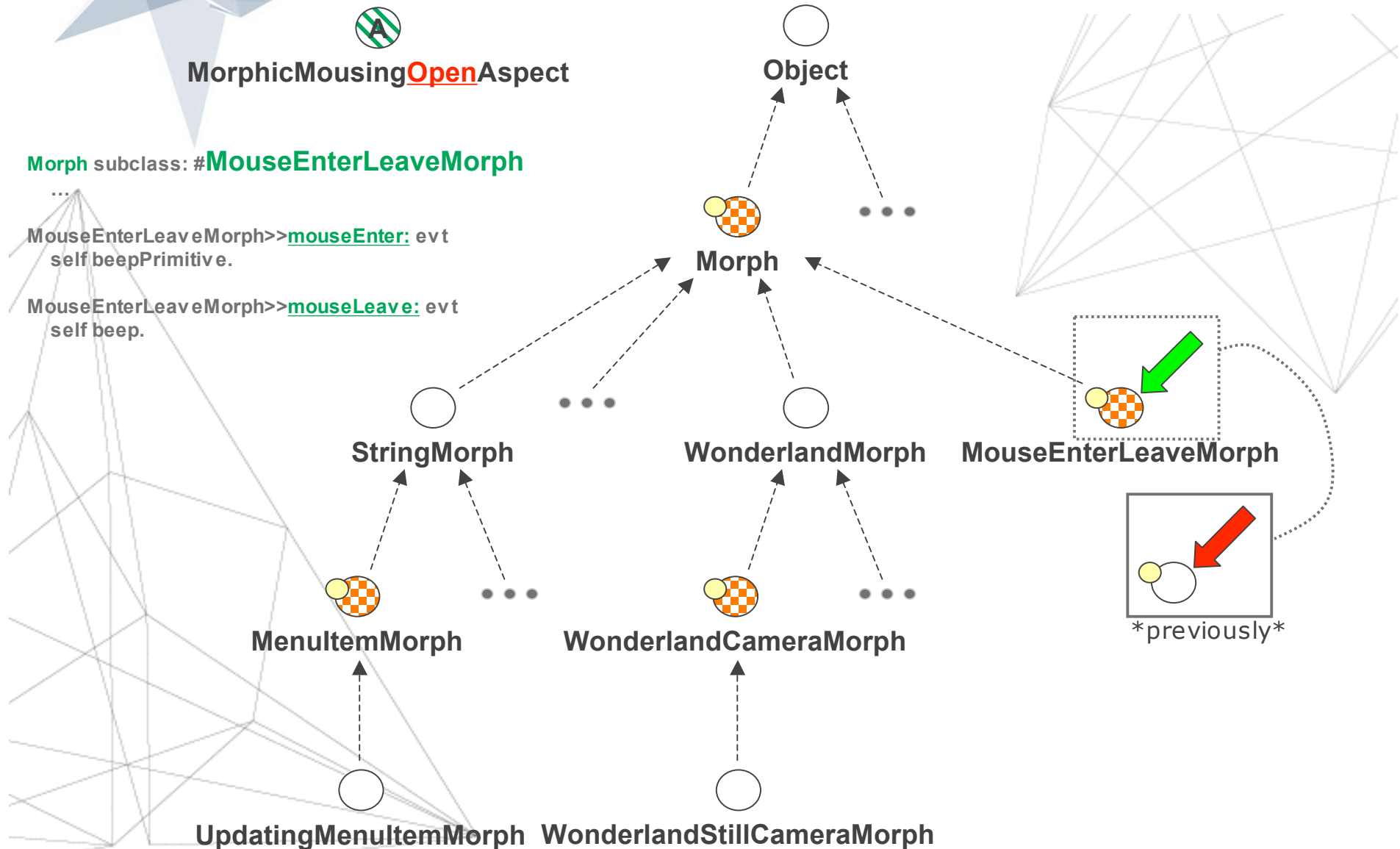


Fig. 21

# Mobile Adventure With Open Aspects (Subtractive Change)

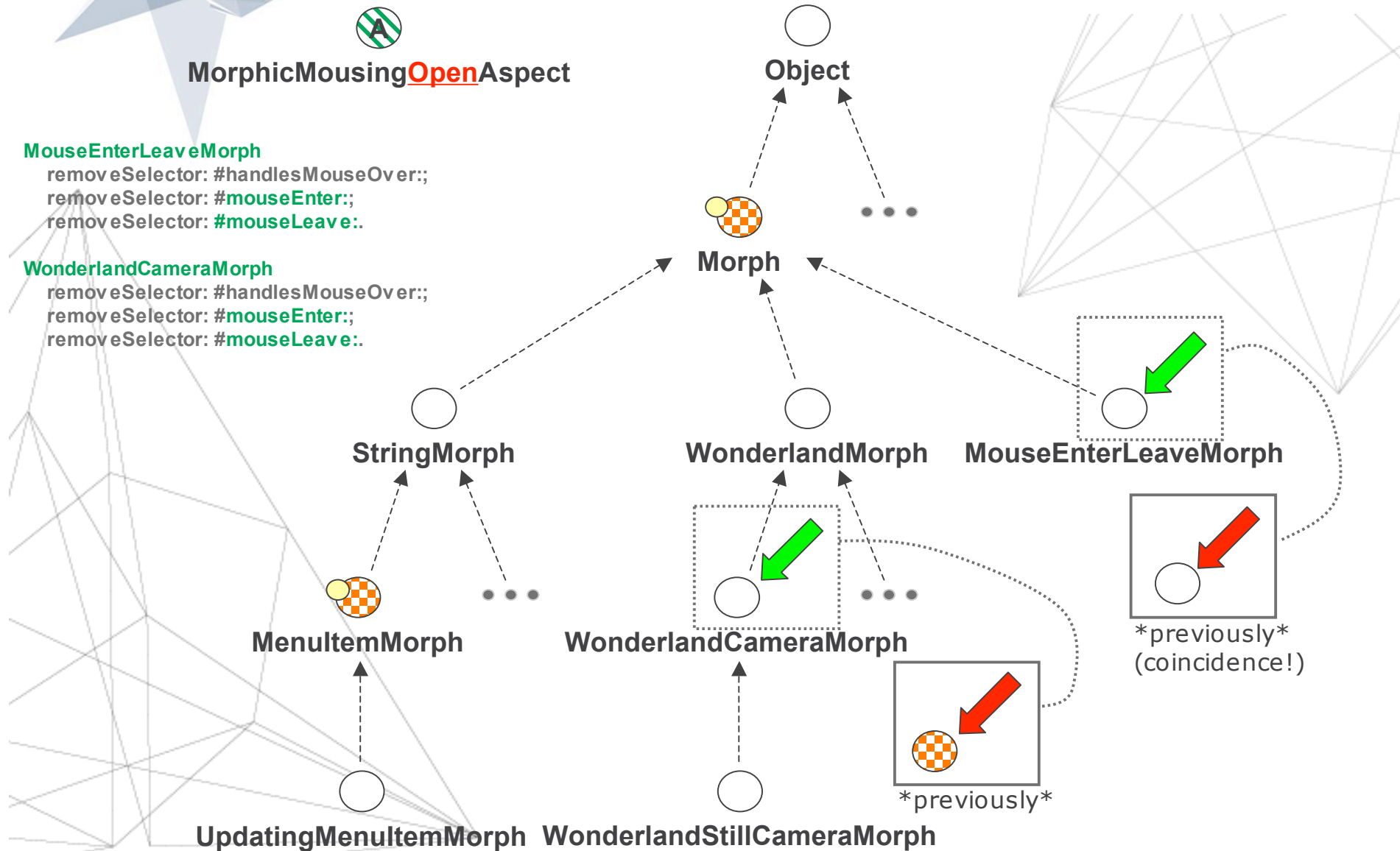


Fig. 23

# Mobile Adventure

## With Open Aspects (Pointcut Reduction)

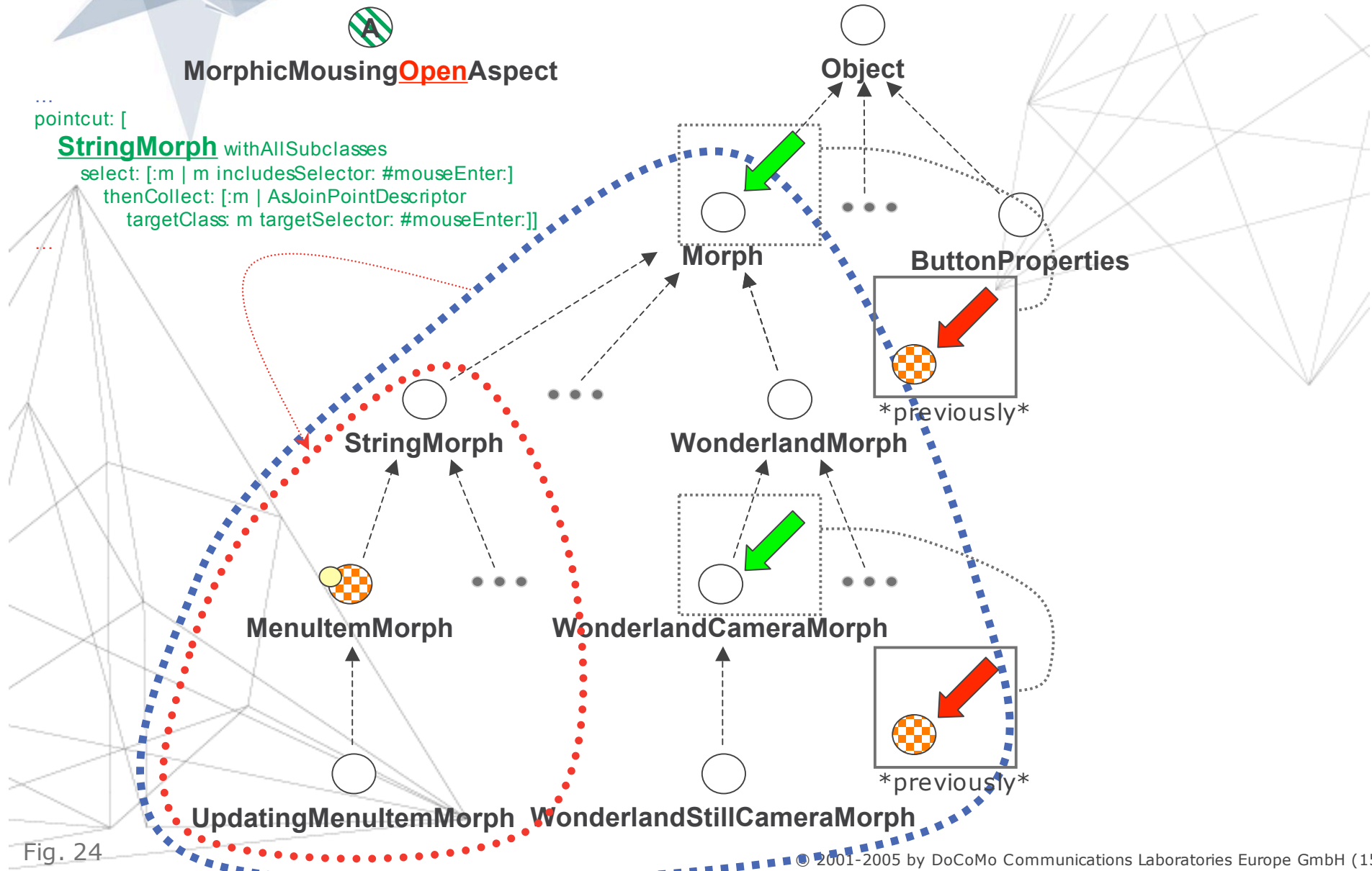


Fig. 24

# Mobile Adventure

## With Open Aspects (Advice Modification)



MorphicMousingOpenAspect

```
...
beforeBlock [:receiver :arguments :aspect :client |
  Logger count increment.
  Transcript show: '*Enter*', arguments first printString]
...
```

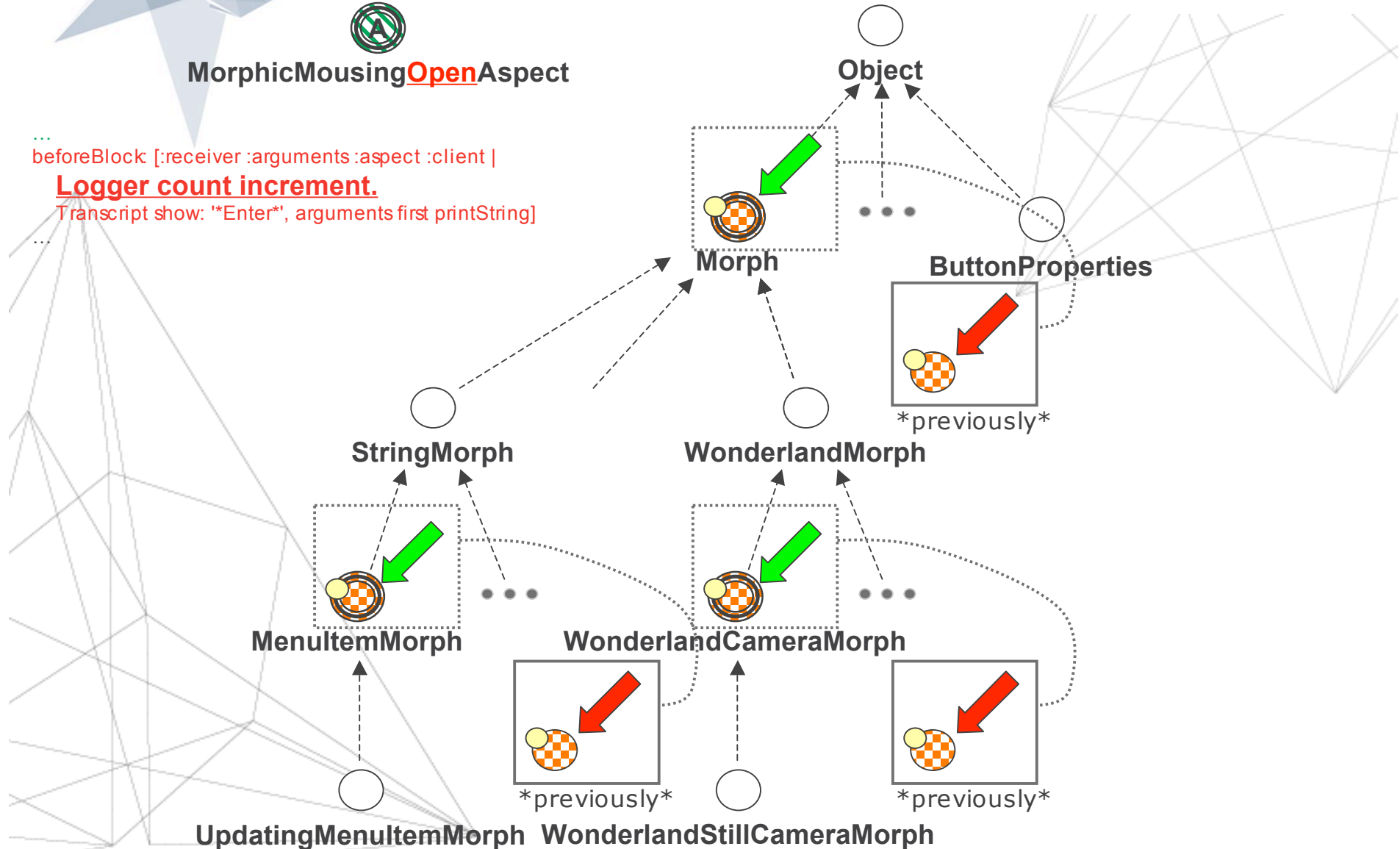


Fig. 25



# Mobile Adventure

## AspectS Weaving

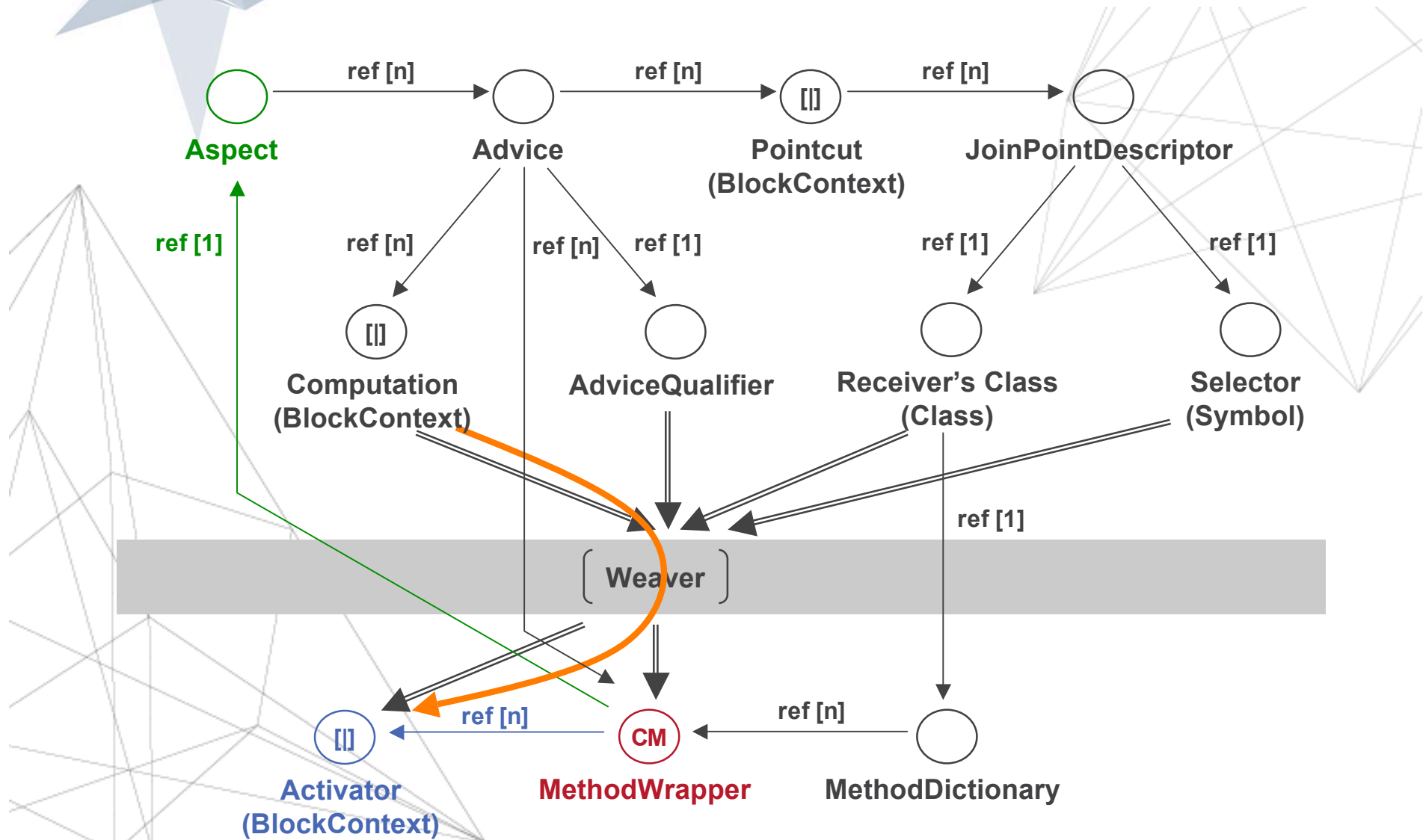


Fig. 16

# Mobile Adventure

## Runtime Structure Extensions

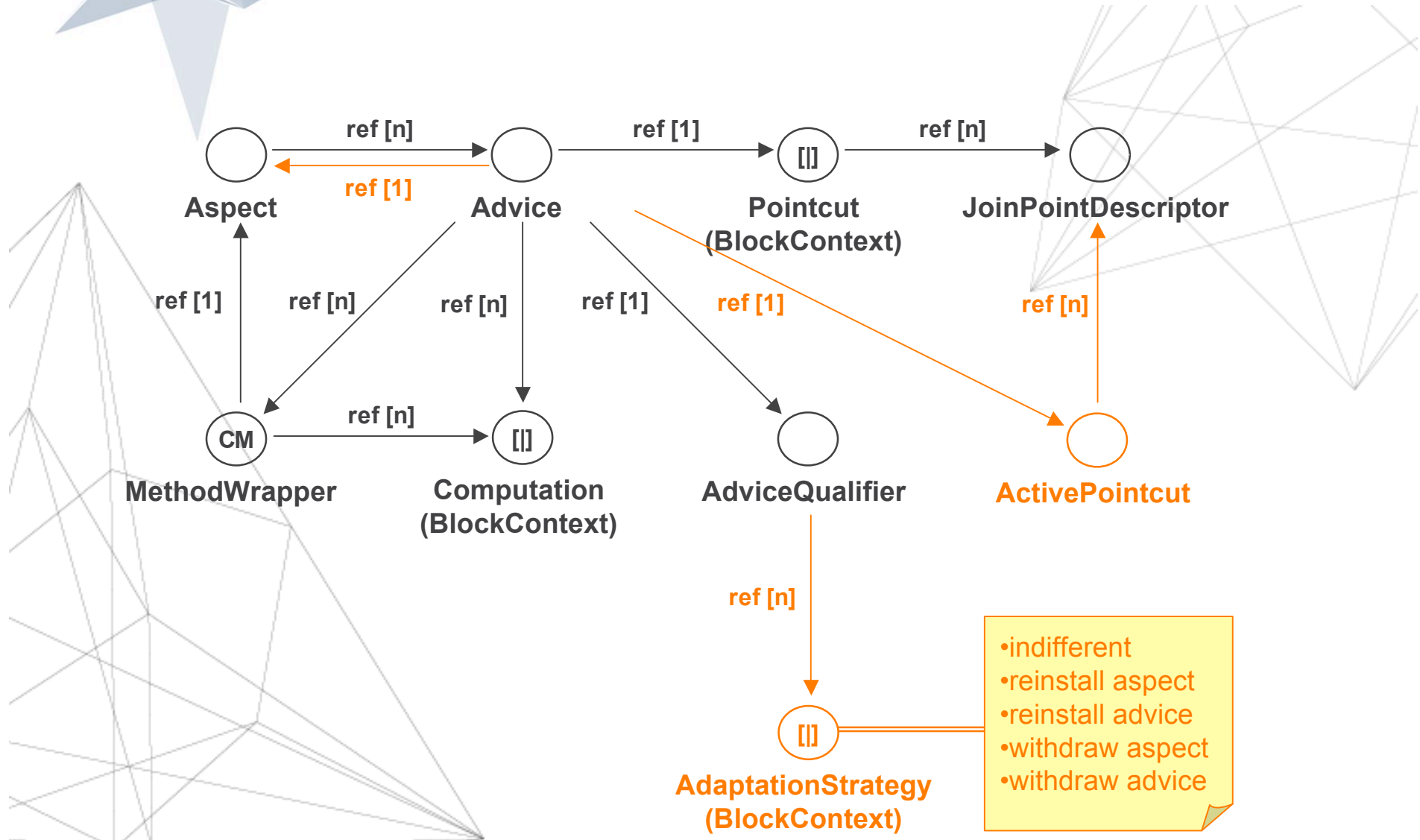


Fig. 17

# Mobile Adventure

## Comparison for Relevance

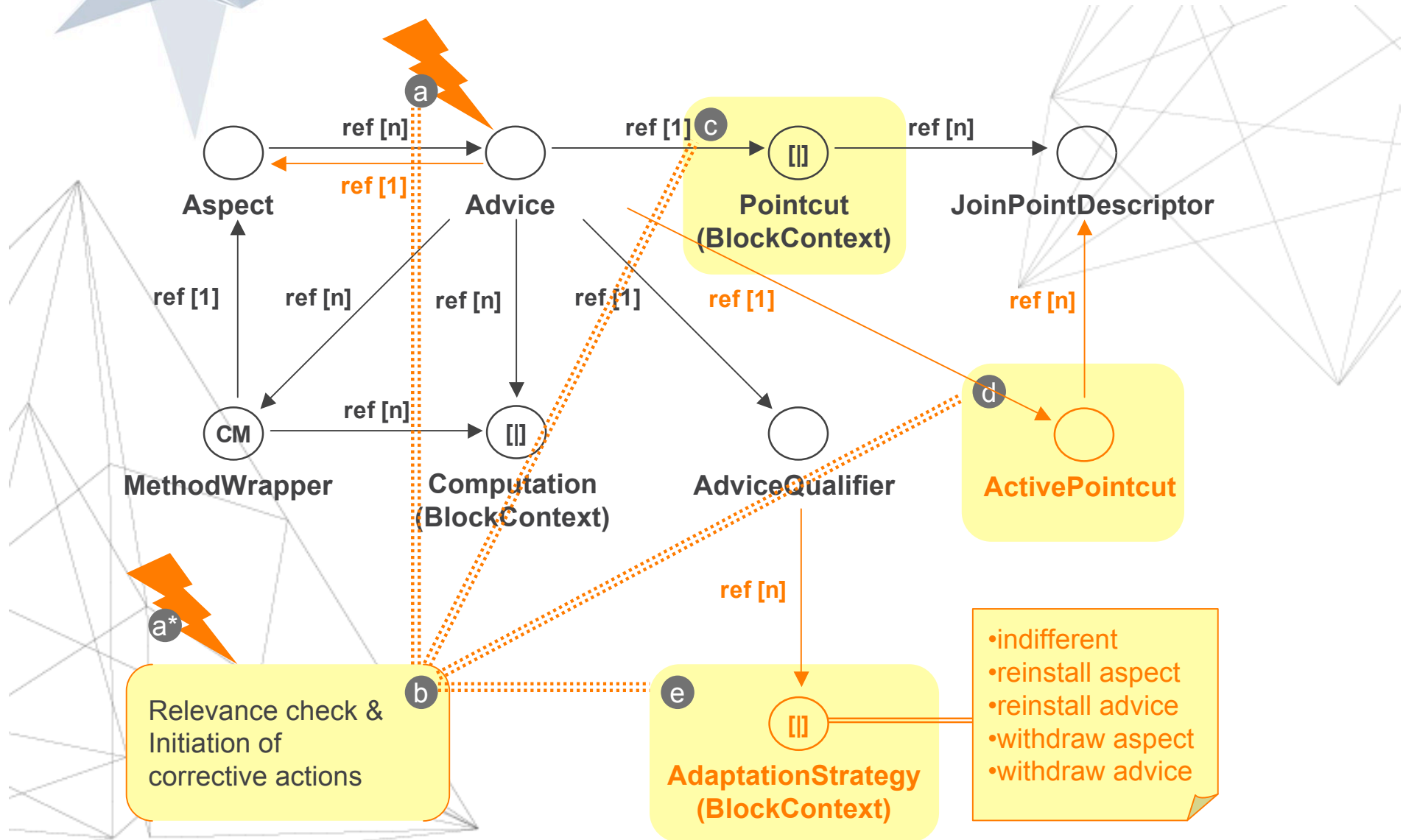
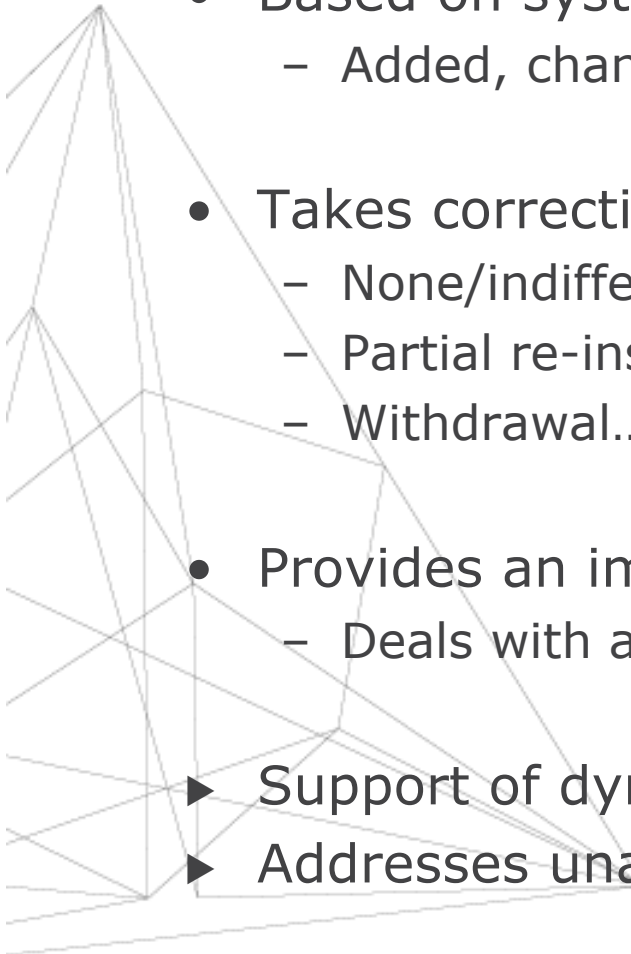
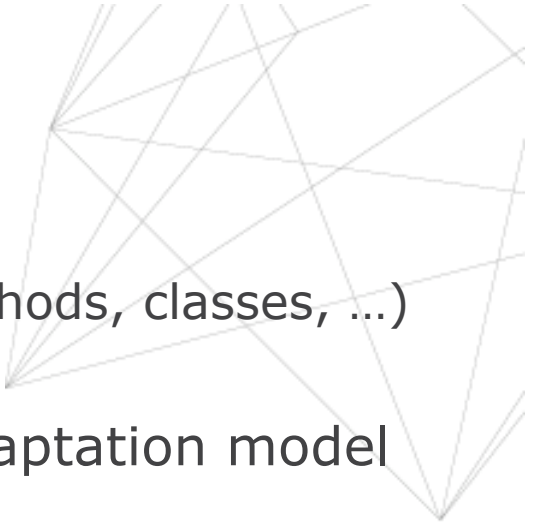


Fig. 18

- 
- 
- Conditional (re-) composition mechanism
  - Based on system change events
    - Added, changed, or removed objects (methods, classes, ...)
  - Takes corrective actions using explicit adaptation model
    - None/indifferent
    - Partial re-install
    - Withdrawal...
  - Provides an improved composer (weaver)
    - Deals with additive and subtractive changes
  - ▶ Support of dynamic AOP in open systems
  - ▶ Addresses unanticipated software evolution



# Open Aspects

Robert Hirschfeld

DoCoMo Euro-Labs

Stefan Hanenberg

University of Duisburg-Essen

ESUG, Brussels, 2005-08-16