Aspects: The pointcut question

Julia Lawall, DIKU, University of Copenhagen

January 26, 2005

Questions

- What can we do with aspects besides logging?
- Are existing pointcut languages expressive enough?

AspectJ pointcuts:

- Joinpoints: method invocation, field references
- Modifiers: within, cflow
- Selectors: return types, parameter lists, wildcards

Issue: are method invocation and field references enough?

Example: Locking is a non-functional concern

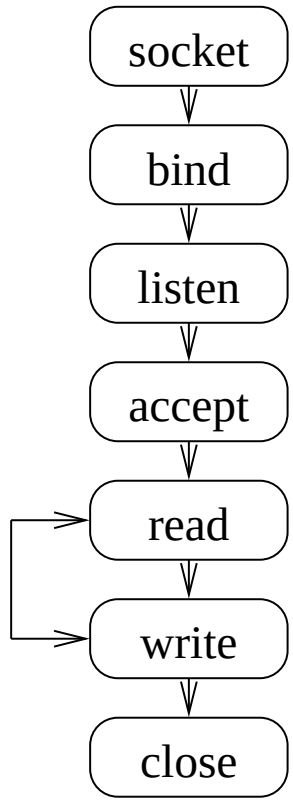
[Lohmann, Spinczyk, Schröder-Preikschat, ACP4IS05]

Locking in Linux (drivers/usb/class/audio.c)

```
spin_lock_irqsave(&as->lock, flags);
if (!(u->flags & FLG_CONNECTED)) {
  spin_unlock_irqrestore(&as->lock, flags);
  return -EIO;
}
if (!(u->flags & FLG_RUNNING)) {
  spin_unlock_irqrestore(&as->lock, flags);
  u->freqn = ((u->dma.srate << 11) + 62) / 125;
  ...
  spin_lock_irqsave(&as->lock, flags);
}
if (u->dma.count >= u->dma.dmasize && !u->dma.mapped) {
  spin_unlock_irqrestore(&as->lock, flags);
  return 0;
}
...
```

Issue: is cflow enough?

Example: Replace the use of TCP by the use of UDP
[Douence, et al., AOSD05]



Need to be able to describe sequences of operations

Issue: other control-flow properties

How to add some code at the earliest or latest point that dominates some other code.

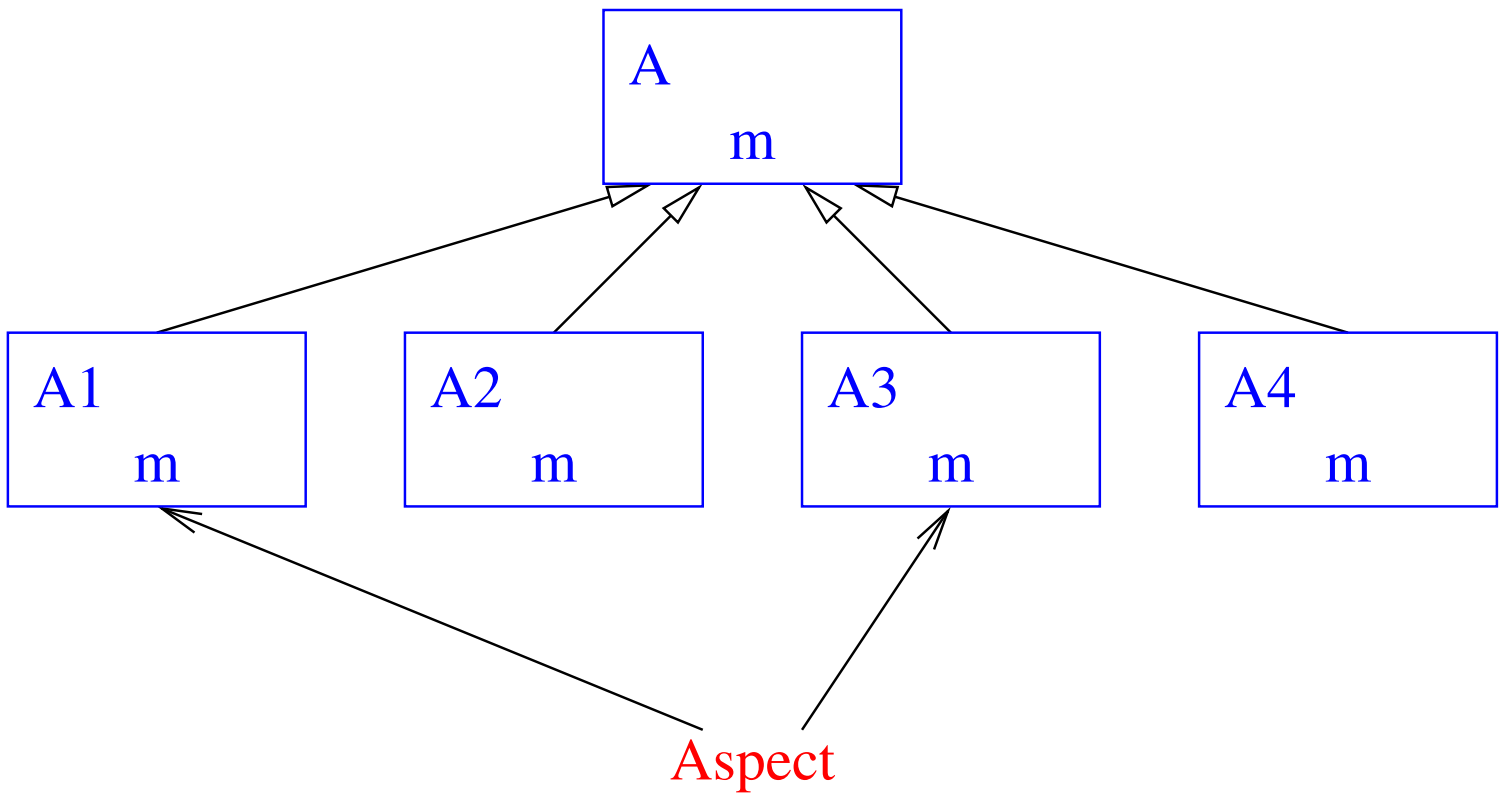
- Relevant in a source-to-source aspect system.

```
if (test) {
  f(1,2,3);
}
else {
  f(4,5,6);
}
```

Issue: what kind of structure is relevant?

Example: Multiple methods override a method, only some should be affected by the aspect

[Tourwé, Brichau, Gybels, AOSD'03 workshop]



Conclusion

- There are many issues in program design that must be taken into account.
- The structuring constraints of aspects are one among many.
- Refactoring a program to coincide with the constraints of a pointcut language is not a good solution.