

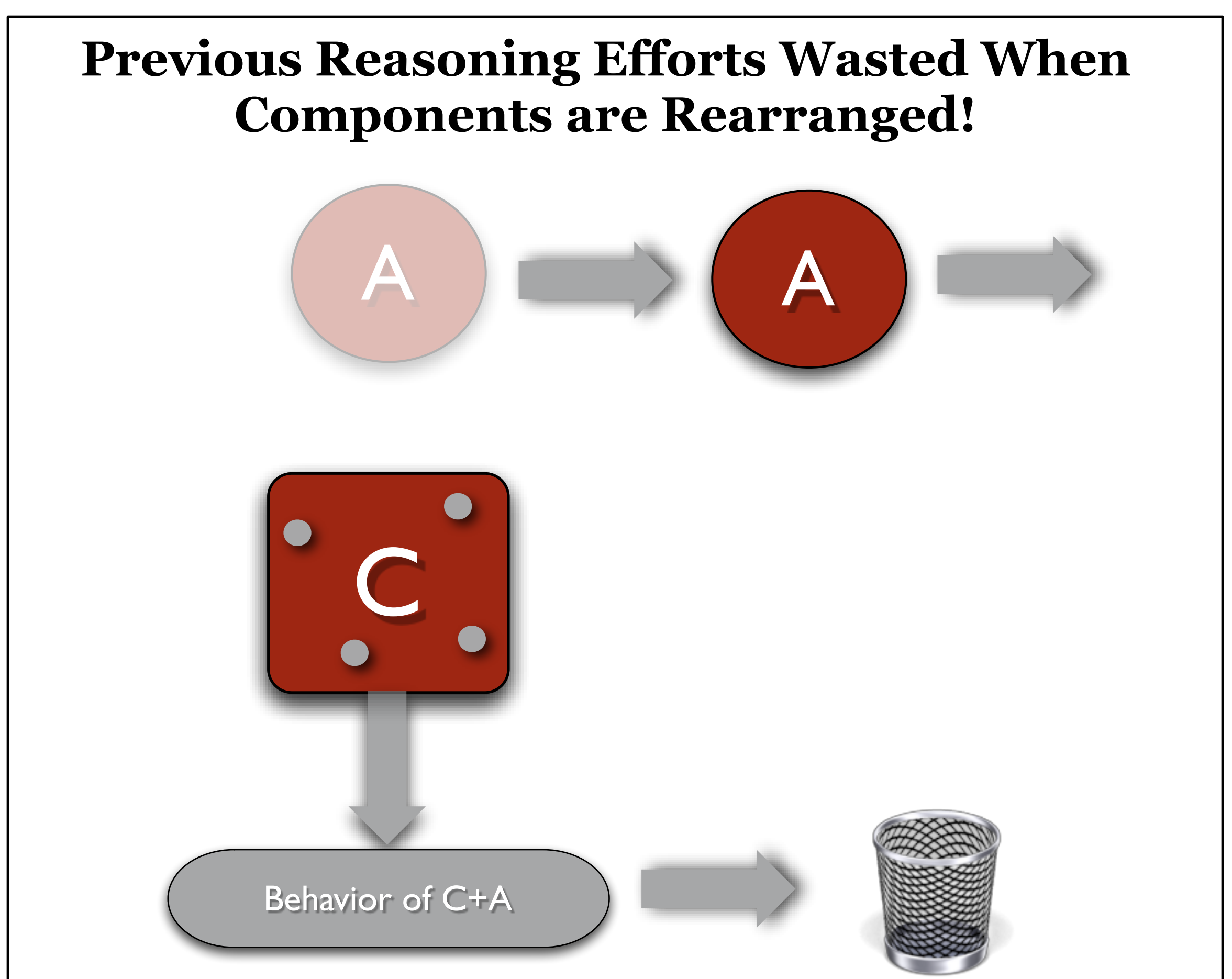
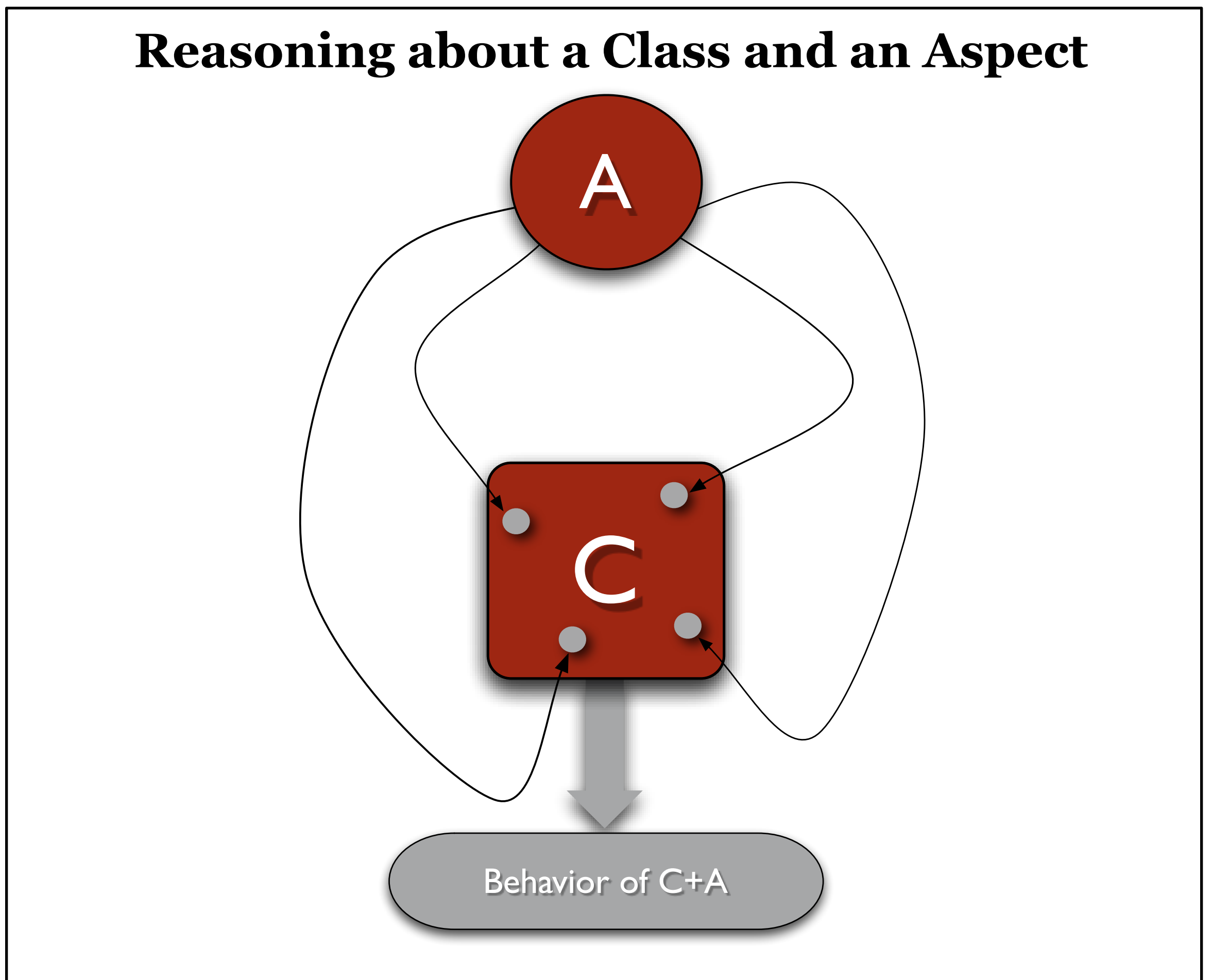
# Enforcing Behavioral Constraints in Evolving Aspect-Oriented Programs

Raffi Khatchadourian<sup>1</sup>, Johan Dovland<sup>2</sup>, and Neelam Soundarajan<sup>1</sup>  
<sup>1</sup>The Ohio State University, USA and <sup>2</sup>University of Oslo, Norway

**Reasoning in Aspect-Oriented Programs (AOP)**

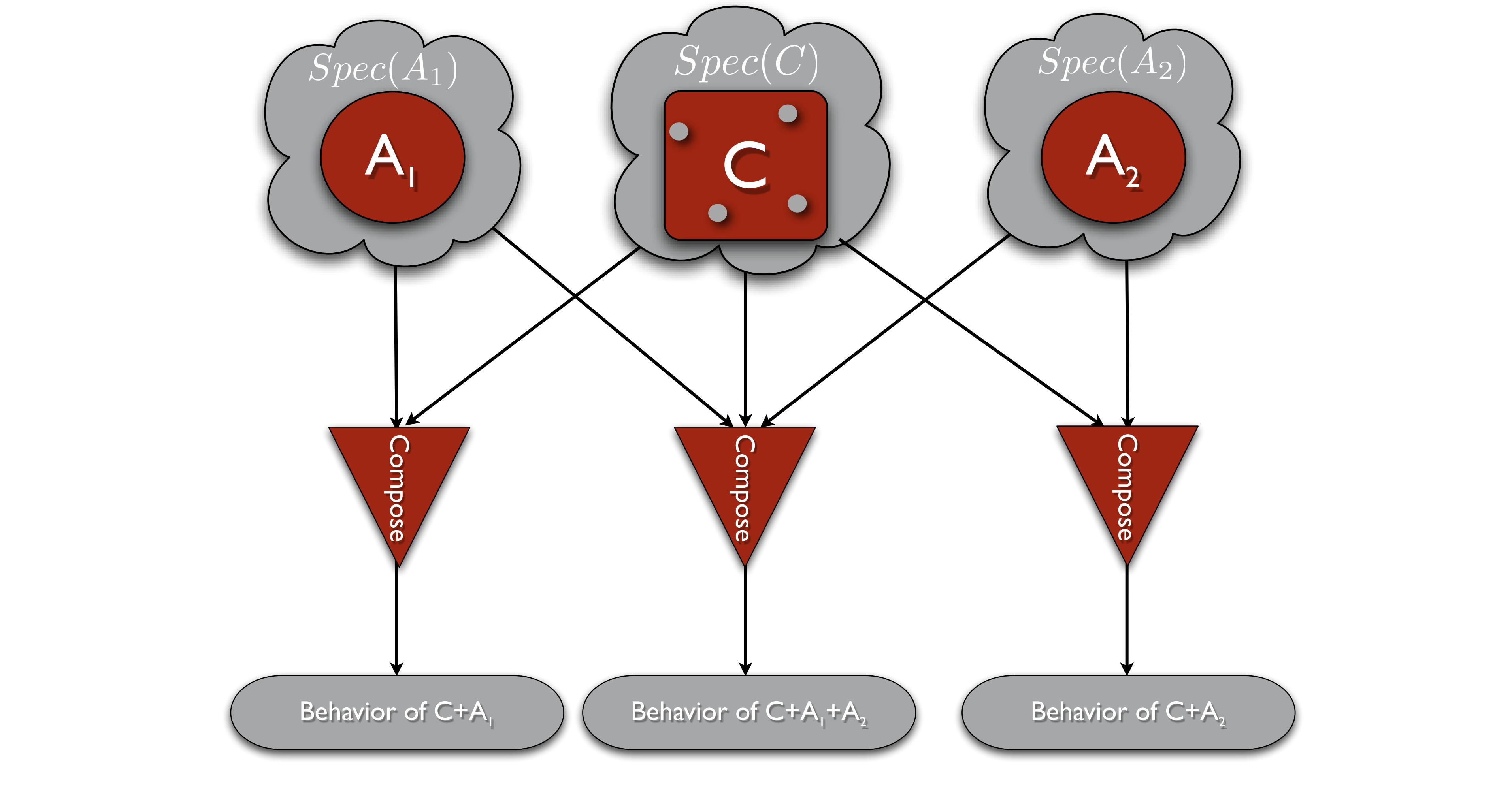
- AOP enables modular implementations of crosscutting concerns.
- Both *formal* and *informal* reasoning about AOP presents unique challenges especially in respect to evolution.
- As components enter, exit, and re-enter software, conclusions about behavior of components may be invalidated.

**“Can AOP facilitate modular reasoning of crosscutting concerns?”**



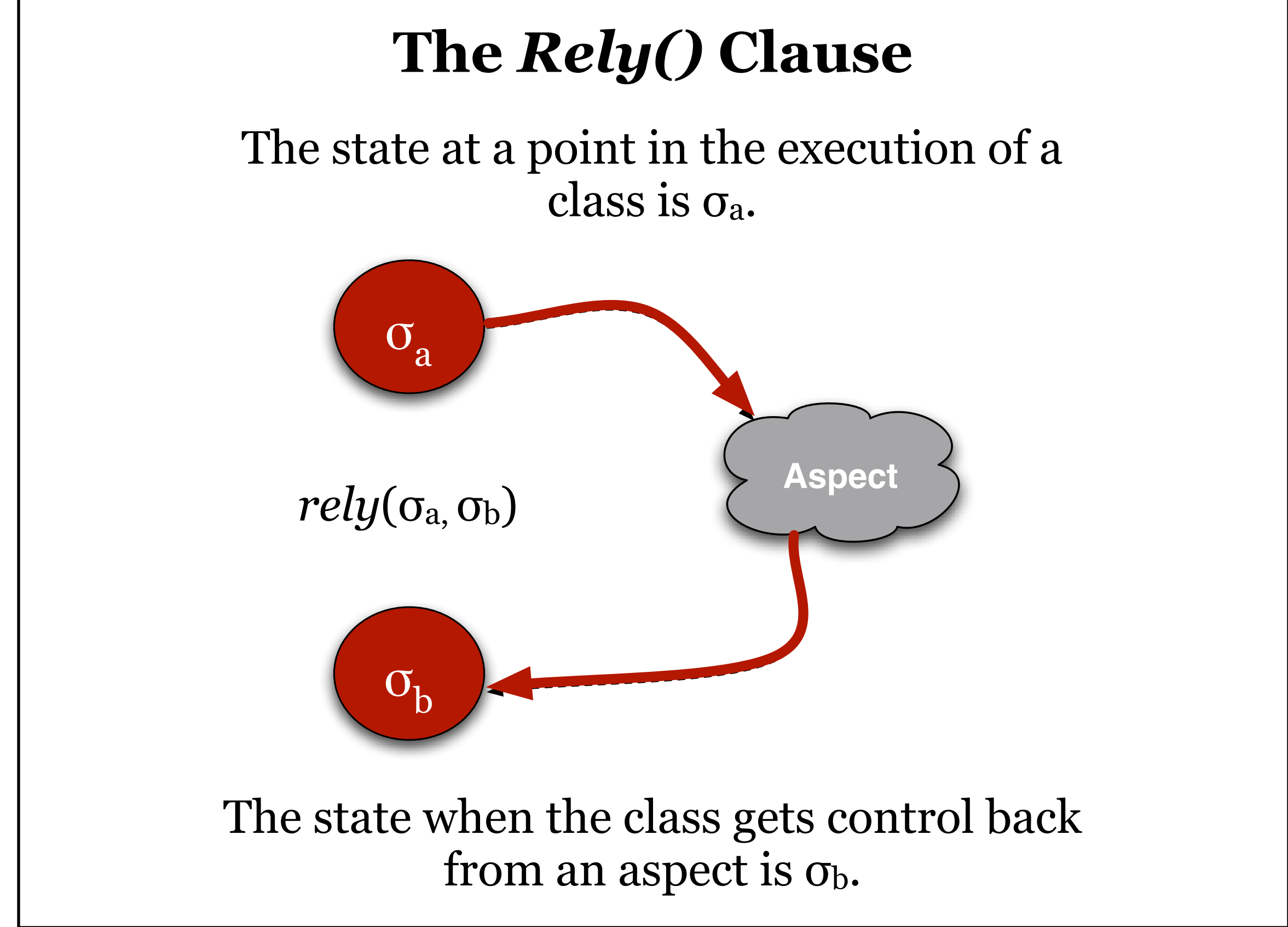
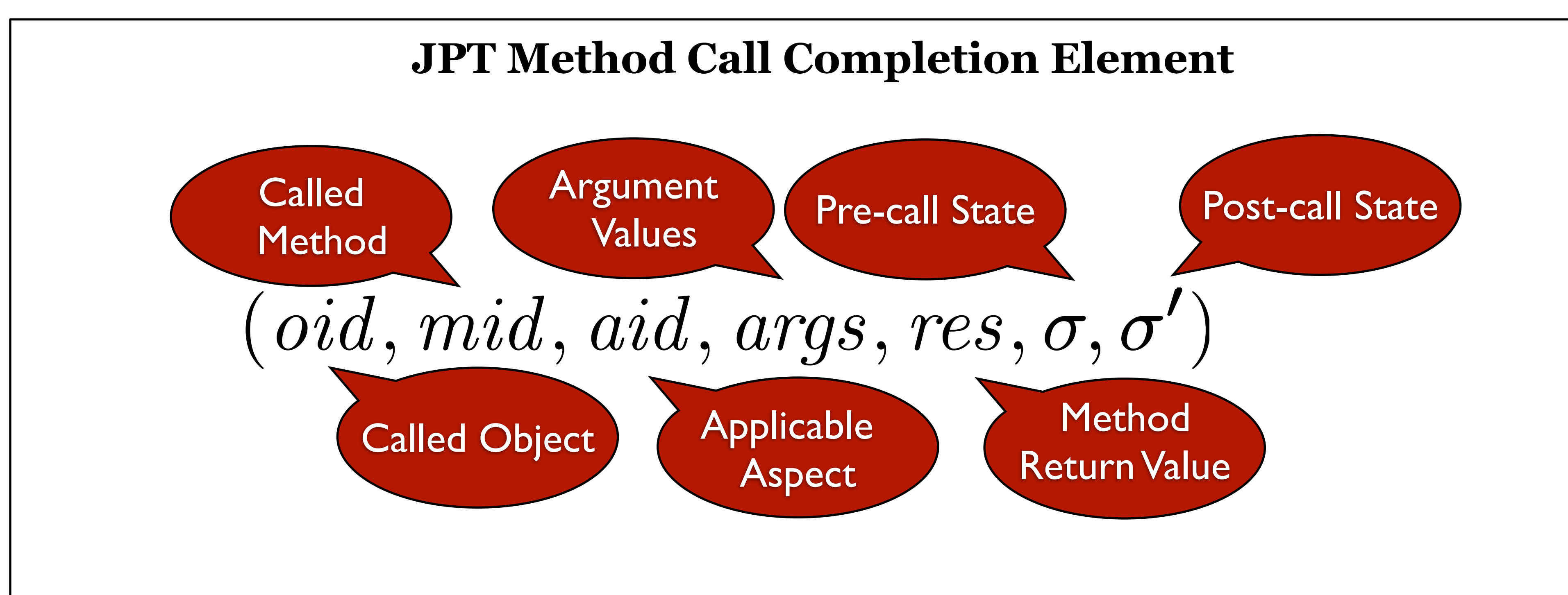
**Composing “Plug-N-Play” Specifications to Derive *Enriched Behavior***

- AO programs inherently enjoy *plug-n-play* capabilities.
- Crosscutting features can be *plugged-in* to *enrich* the behavior of advised components.
- Likewise, can we specify components so that we can derive their behaviors in a similar fashion?



**Join Point Traces (JPTs): A Trace-based Approach to AO Reasoning**

- Specification variable to track the *flow-of-control* through join points.
- Used to capture actions and resulting values taken by possible advice.



**Rule for Method Specification**

$$\frac{pre \wedge [\lambda\tau = \langle (inv, C.m) \rangle] \Rightarrow p \quad \{p\}S\{q\} \quad q \Rightarrow guar() \quad [q \wedge rely(\sigma, \sigma')] \Rightarrow post[\lambda\tau \leftarrow \lambda\tau \hat{\ } (this, C.m, ?, args, res, \sigma, \sigma'), \sigma \leftarrow \sigma']}{C.m :: \langle pre, post, guar(), rely() \rangle}$$

**Rule for Method Calls**

$$\frac{p \Rightarrow C.m.pre[pars/args] \quad C.m.post \Rightarrow q[\lambda\tau 1/\lambda\tau 1 \hat{\ } \lambda\tau 2, args/pars]}{\{p\} ob.m(args) \{q\}}$$

**Rule for Aspect Application**

$$\frac{\{ guar(\sigma) \wedge ap \} A_{adv} \{ rely[\sigma/\sigma @ pre, \sigma'/\sigma] \wedge aq \}}{C.m :: \langle pre, post, guar, rely \rangle \quad \{ pre \wedge ap \} C.m() + A \{ post \wedge aq \}}$$