aspect OutcomeVotable depends on OutcomeAware, Terminatable

structural view

IOutcomeVotableParticipant

OutcomeVotableContext

~ OutcomeVotableContext create()
+ pauseContext()
+ continueContext()
+ terminateContext(IOutcomeVotableParticipant)
+ registerVote(Outcome, IOutcomeControllingParticipant)
+ boolean alreadyVoted(IOutcomeVotableParticipant)
- Outcome computeOutcome()
- boolean isOutcomeDecided()

votes 1 1 IOutcomeVotableParticipant, Outcome

Map

~ Map<OutcomeVotableParticipant, Outcome> create()
~ put(IOutcomeVotableParticipant, Outcome)
~ boolean containsKey(IOutcomeVotableParticipant)
~ Collection<Outcome> values()

IOutcomeVotableParticipant

+ leaveContext()
+ voteAndLeaveContext(Outcome)

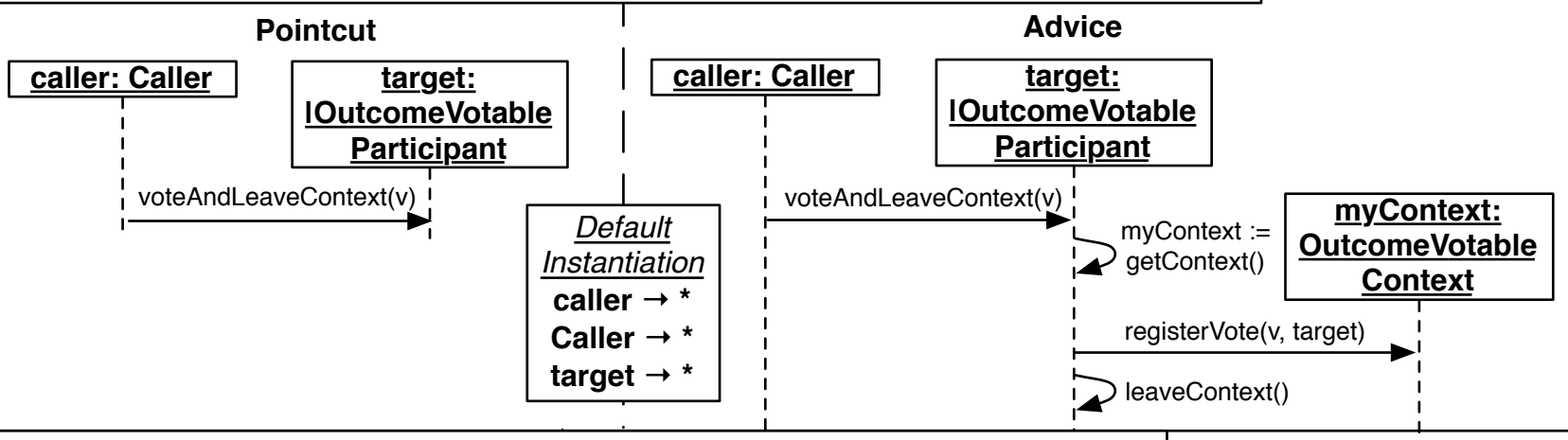
OutcomeAware instantiation
IOutcomeControllingParticipant → IOutcomeVotableParticipant

Terminatable instantiation
ITerminatableParticipant → IOutcomeVotableParticipant

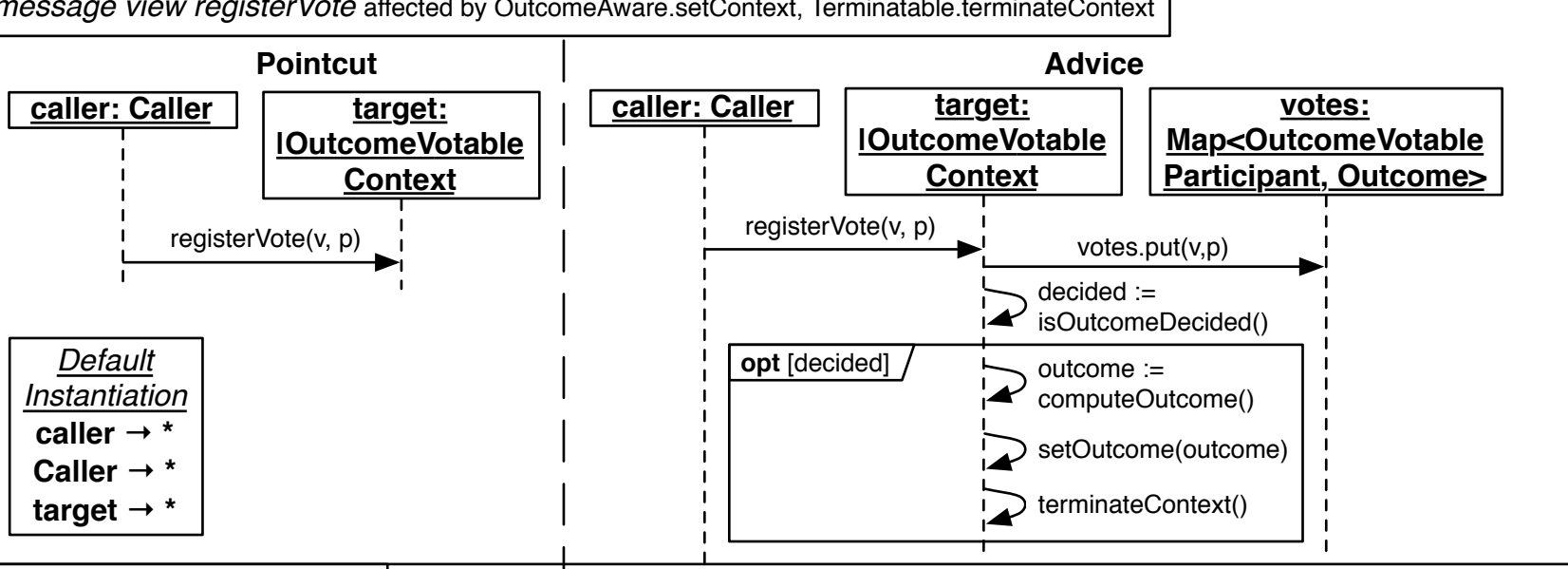
Terminatable binding
OutcomeVotableContext → TerminatableContext
leaveContext → leaveContext
pauseContext → pauseContext
continueContext → continueContext
terminateContext → terminateContext

OutcomeAware binding
OutcomeVotableContext → ContextWithOutcome

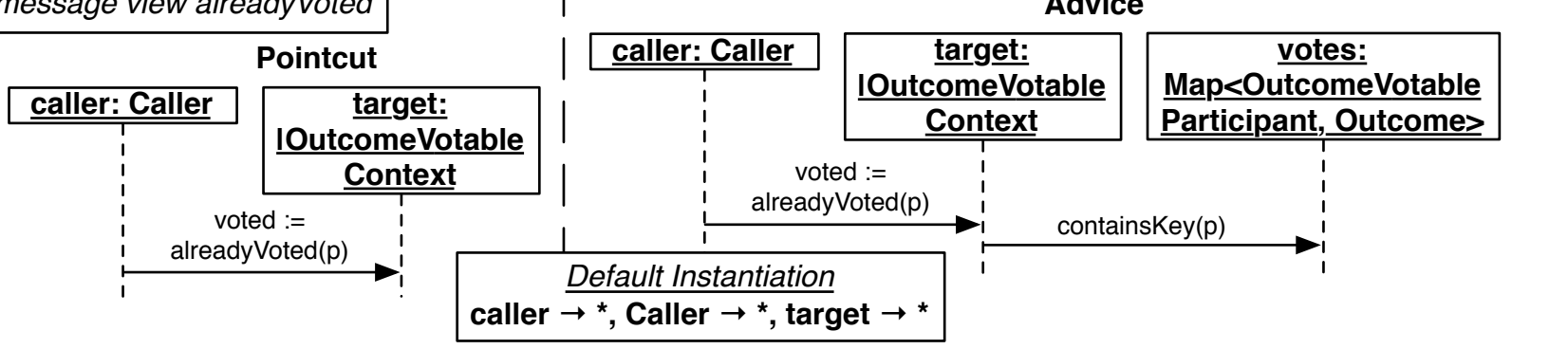
message view voteAndLeaveContext affected by OutcomeAware.getContext, Terminatable.leaveContext



message view registerVote affected by OutcomeAware.setContext, Terminatable.terminateContext



message view alreadyVoted



message view computeOutcome affected by OutcomeAware.BinaryOutcome.create