aspect Recovering depends OutcomeAware, Checkpointing

structural view

IRecoveringParticipant	**IRecoverable**	**RecoveringContext**
+ createAndEnterContext()	+ * m(..)	- contextCompleted()
+ voteAndLeaveContext()		

IRecoveringParticipant
IRecoverable

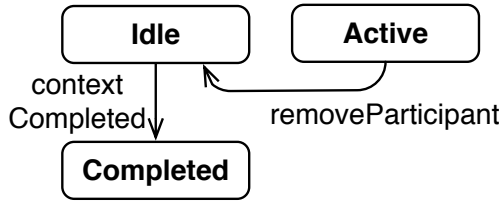*Checkpointing instantiation*
**Checkpointing.CheckpointingContext →
RecoveringContext
Checkpointing.ICheckpointingParticipant →
IRecoveringParticipant
Checkpointing.ICheckpointed →
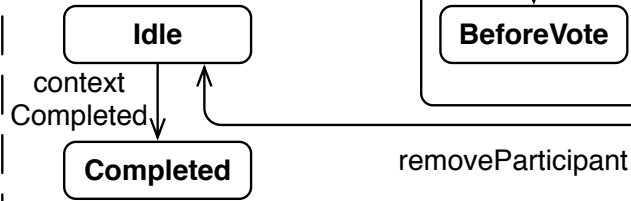IRecoverable**

OutcomeAware instantiation
**OutcomeAware.ContextWithOutcome →
RecoveringContext
OutcomeAware.IOutcomeControllingParticipant →
IRecoveringParticipant**

---

state view RecoveringContext depends on CheckpointingContext, ContextWithOutcome

Pointcut

Idle   **Active**

context
Completed ↓   removeParticipant

Completed

Advice

Idle

context
Completed ↓   removeParticipant

Completed

Active

● → **BeforeVote**   **AfterVote**

Checkpointing.CheckpointingContext instantiation
**CheckpointingContext.Active → BeforeVote
CheckpointingContext.Idle → Idle
CheckpointingContext.Completed → Completed
CheckpointingContext.RestoreAllowed →
Completed**

Context binding
**Active → Context.Active
Idle → Context.Idle
Completed → Context.Completed**

OutcomeAware.ContextWithOutcome instantiation
**ContextWithOutcome.Undecided → BeforeVote
ContextWithOutcome.Decided → AfterVote**

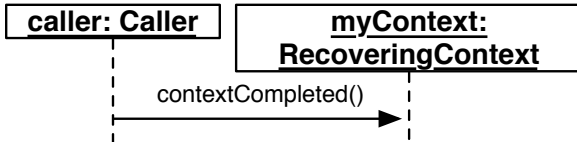*state view IRecoveringParticipant* is IOutcomeControllingParticipant

IIdle, IWorking

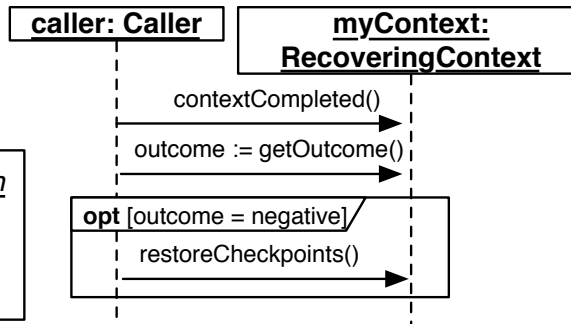*state view IRecoverable* is ICheckpointed

---

message view voteAndLeaveContext is OutcomeAware.voteAndLeaveContext binds contextCompleted

---

message view contextCompleted depends on Checkpointing, OutcomeAware

Pointcut

caller: Caller   **myContext:
RecoveringContext**

contextCompleted()

Checkpointing.restoreCheckpoints instantiation
**restoreCheckpoints.target → myContext
restoreCheckpoints.Caller → Caller
restoreCheckpoints.caller → caller**

Checkpointing.leaveContext binding
**target → Checkpointing.leaveContext.myContext
Caller → Checkpointing.CheckpointingContext
caller → Checkpointing.leaveContext.myContext**

Advice

caller: Caller   **myContext:
RecoveringContext**

contextCompleted()

outcome := getOutcome()

opt [outcome = negative]
restoreCheckpoints()

| message view *prepareRecovery* is Checkpointing.checkpointMethod | |
| message view *createAndEnterContext* is OutcomeAware.createAndEnterContext |