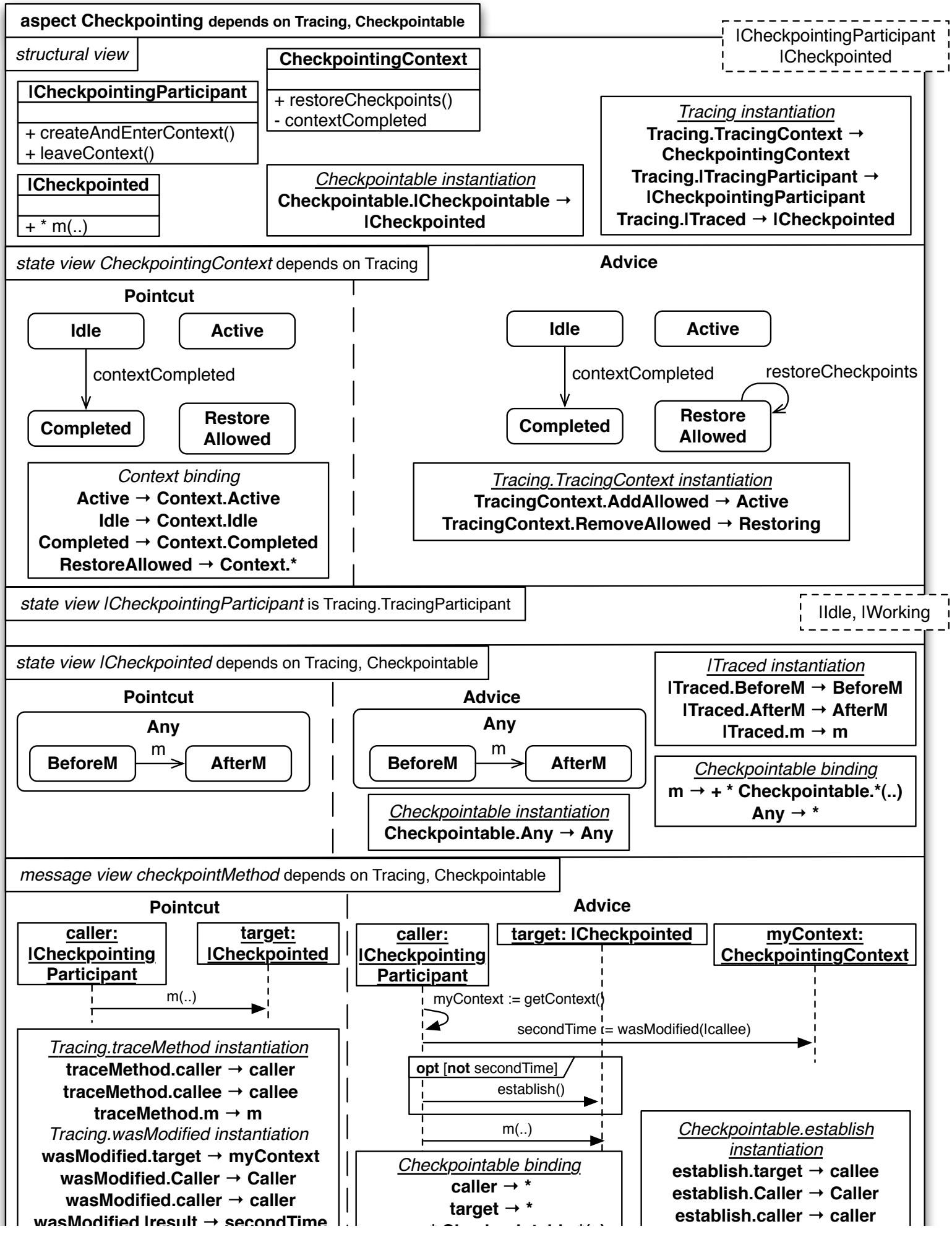
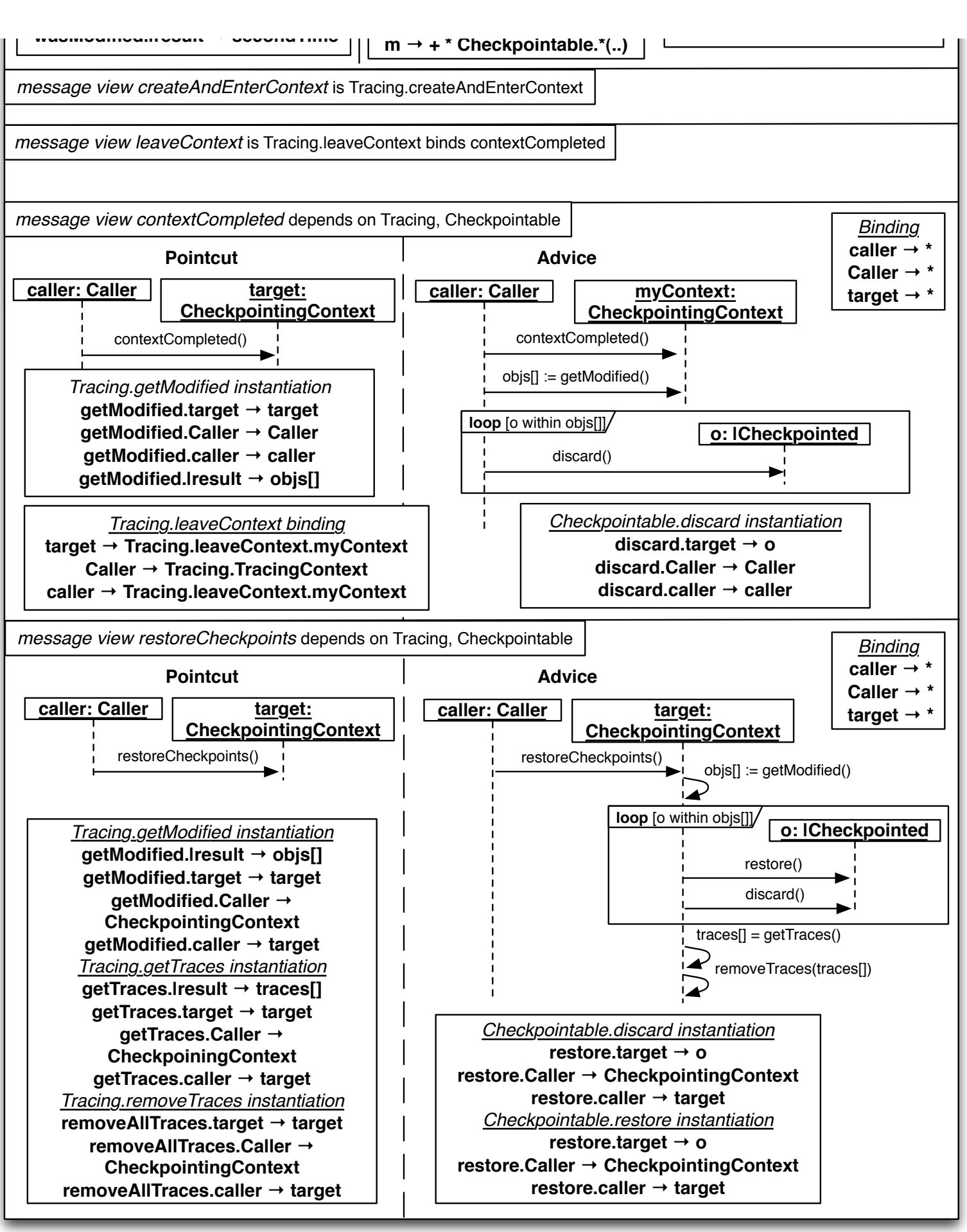

aspect Checkpointing depends on Tracing, Checkpointable
ICheckpointingParticipant
ICheckpointed
structural view
CheckpointingContext
+ restoreCheckpoints()
- contextCompleted
ICheckpointingParticipant
+ createAndEnterContext()
+ leaveContext()
ICheckpointed
+ * m(..)
Checkpointable instantiation
Checkpointable.ICheckpointable → ICheckpointed
Tracing instantiation
Tracing.TracingContext → CheckpointingContext
Tracing.ITracingParticipant → ICheckpointingParticipant
Tracing.ITraced → ICheckpointed
state view CheckpointingContext depends on Tracing
Pointcut
Idle
Active
contextCompleted
Completed
Restore Allowed
Context binding
Active → Context.Active
Idle → Context.Idle
Completed → Context.Completed
RestoreAllowed → Context.*
Advice
Idle
Active
contextCompleted
restoreCheckpoints
Completed
Restore Allowed
Tracing.TracingContext instantiation
TracingContext.AddAllowed → Active
TracingContext.RemoveAllowed → Restoring
state view ICheckpointingParticipant is Tracing.TracingParticipant
IIdle, IWorking
state view ICheckpointed depends on Tracing, Checkpointable
Pointcut
Any
m
BeforeM
AfterM
Advice
Any
m
BeforeM
AfterM
Checkpointable instantiation
Checkpointable.Any → Any
ITraced instantiation
ITraced.BeforeM → BeforeM
ITraced.AfterM → AfterM
ITraced.m → m
Checkpointable binding
m → + * Checkpointable.*(..)
Any → *
message view checkpointMethod depends on Tracing, Checkpointable
Pointcut
caller: ICheckpointing Participant
target: ICheckpointed
m(..)
Tracing.traceMethod instantiation
traceMethod.caller → caller
traceMethod.callee → callee
traceMethod.m → m
Tracing.wasModified instantiation
wasModified.target → myContext
wasModified.Caller → Caller
wasModified.caller → caller
wasModified.result → secondTime
Advice
caller: ICheckpointing Participant
target: ICheckpointed
myContext: CheckpointingContext
myContext := getContext()
secondTime := wasModified(Icallee)
opt [not secondTime]
establish()
m(..)
Checkpointable binding
caller → *
target → *
Checkpointable.establish instantiation
establish.target → callee
establish.Caller → Caller
establish.caller → caller


m → + * Checkpointable.*(..)
message view createAndEnterContext is Tracing.createAndEnterContext
message view leaveContext is Tracing.leaveContext binds contextCompleted
message view contextCompleted depends on Tracing, Checkpointable
Binding
caller → *
Caller → *
target → *
Pointcut
Advice
caller: Caller
target: CheckpointingContext
myContext: CheckpointingContext
contextCompleted()
objs[] := getModified()
loop [o within objs[]]
o: ICheckpointed
discard()
Tracing.getModified instantiation
getModified.target → target
getModified.Caller → Caller
getModified.caller → caller
getModified.lresult → objs[]
Tracing.leaveContext binding
target → Tracing.leaveContext.myContext
Caller → Tracing.TracingContext
caller → Tracing.leaveContext.myContext
Checkpointable.discard instantiation
discard.target → o
discard.Caller → Caller
discard.caller → caller
message view restoreCheckpoints depends on Tracing, Checkpointable
restoreCheckpoints()
restore()
traces[] = getTraces()
removeTraces(traces[])
Tracing.getModified instantiation
getModified.lresult → objs[]
getModified.target → target
getModified.Caller → CheckpointingContext
getModified.caller → target
Tracing.getTraces instantiation
getTraces.lresult → traces[]
getTraces.target → target
getTraces.Caller → CheckpoiningContext
getTraces.caller → target
Tracing.removeTraces instantiation
removeAllTraces.target → target
removeAllTraces.Caller → CheckpointingContext
removeAllTraces.caller → target
Checkpointable.discard instantiation
restore.target → o
restore.Caller → CheckpointingContext
restore.caller → target
Checkpointable.restore instantiation
restore.target → o
restore.Caller → CheckpointingContext
restore.caller → target