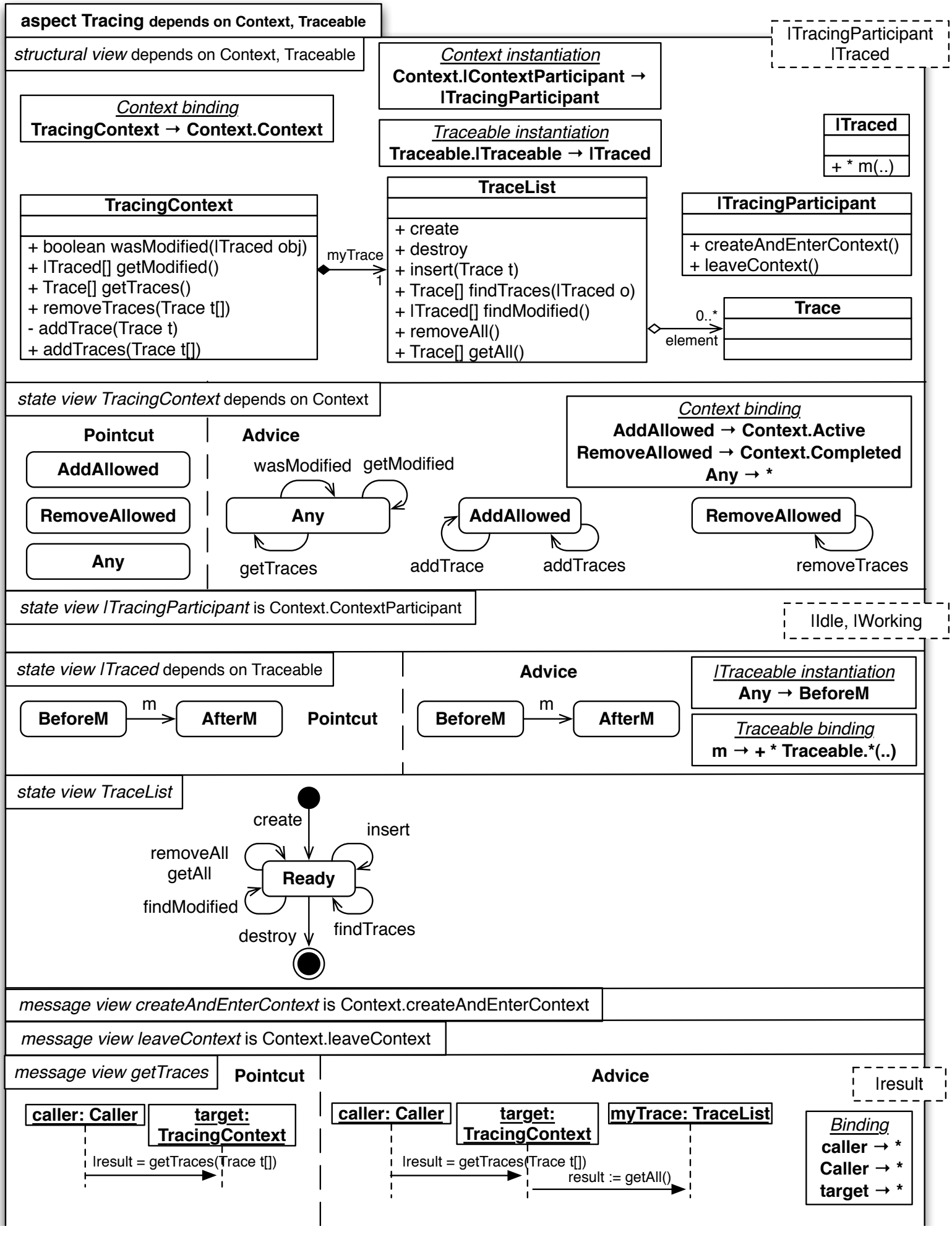
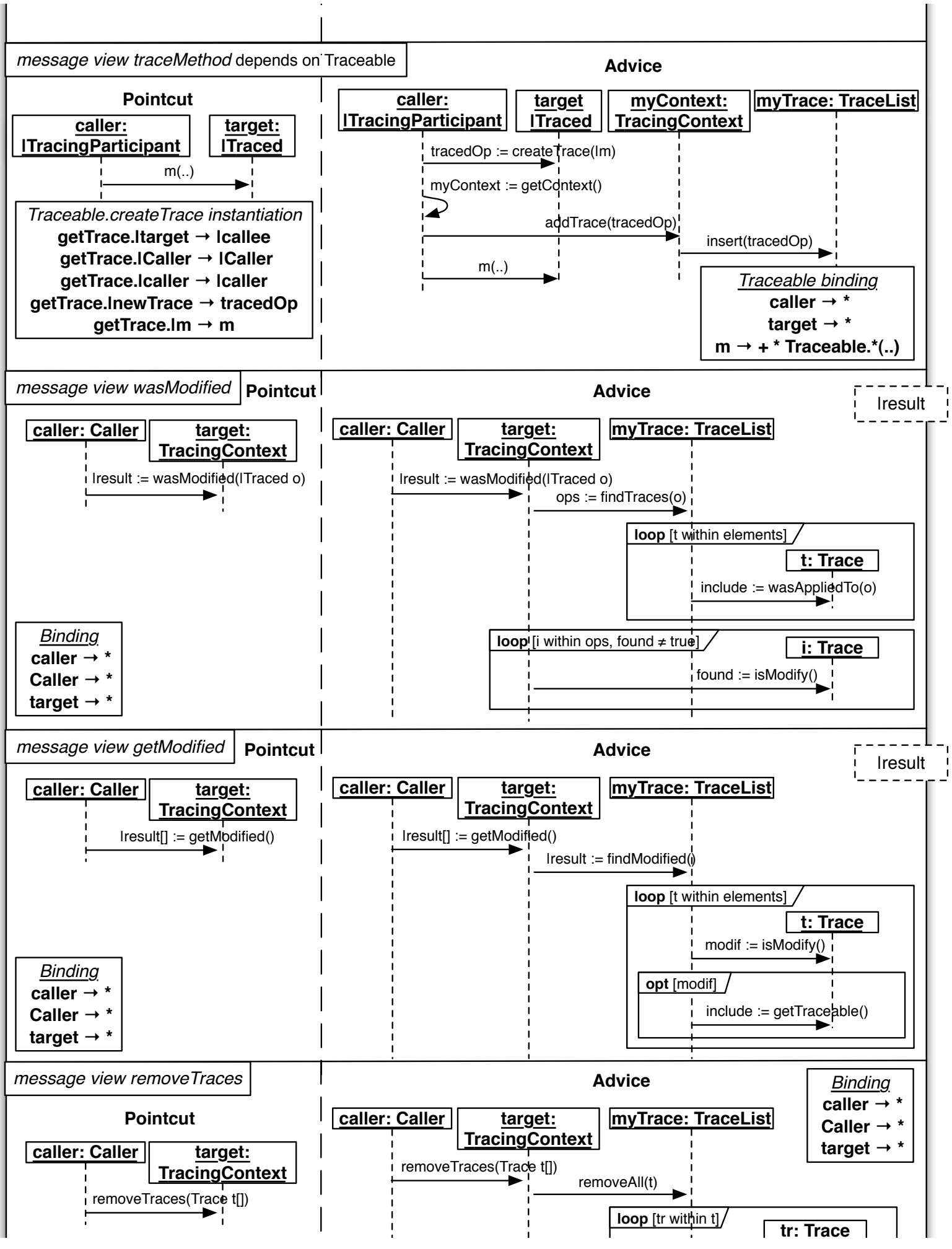
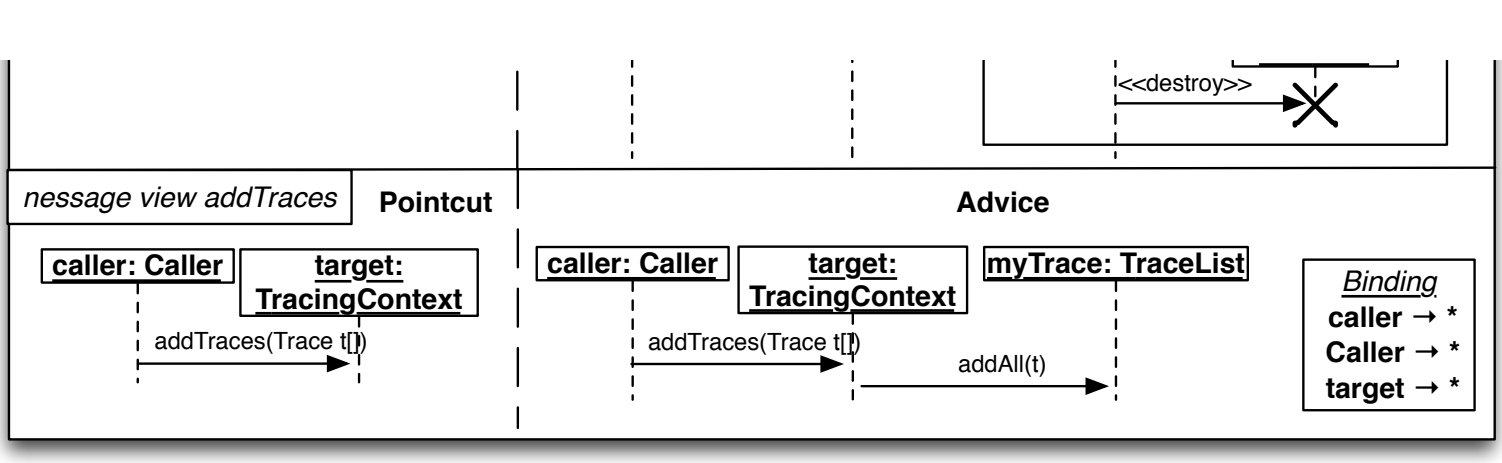

aspect Tracing depends on Context, Traceable
ITracingParticipant
ITraced
structural view depends on Context, Traceable
Context instantiation
Context.IContextParticipant → ITracingParticipant
Context binding
TracingContext → Context.Context
Traceable instantiation
Traceable.ITraceable → ITraced
ITraced
+ * m(..)
TracingContext
+ boolean wasModified(ITraced obj)
+ ITraced[] getModified()
+ Trace[] getTraces()
+ removeTraces(Trace t[])
- addTrace(Trace t)
+ addTraces(Trace t[])
myTrace
1
TraceList
+ create
+ destroy
+ insert(Trace t)
+ Trace[] findTraces(ITraced o)
+ ITraced[] findModified()
+ removeAll()
+ Trace[] getAll()
ITracingParticipant
+ createAndEnterContext()
+ leaveContext()
0..*
element
Trace
state view TracingContext depends on Context
Pointcut
AddAllowed
RemoveAllowed
Any
Advice
wasModified
getModified
Any
getTraces
AddAllowed
addTrace
addTraces
RemoveAllowed
removeTraces
Context binding
AddAllowed → Context.Active
RemoveAllowed → Context.Completed
Any → *
state view ITracingParticipant is Context.ContextParticipant
IIdle, IWorking
state view ITraced depends on Traceable
BeforeM
m
AfterM
Pointcut
Advice
BeforeM
m
AfterM
ITraceable instantiation
Any → BeforeM
Traceable binding
m → + * Traceable.*(..)
state view TraceList
create
insert
removeAll
getAll
Ready
findModified
findTraces
destroy
message view createAndEnterContext is Context.createAndEnterContext
message view leaveContext is Context.leaveContext
message view getTraces
Pointcut
caller: Caller
target: TracingContext
Iresult = getTraces(Trace t[])
Advice
caller: Caller
target: TracingContext
myTrace: TraceList
Iresult = getTraces(Trace t[])
result := getAll()
Iresult
Binding
caller → *
Caller → *
target → *


message view traceMethod depends on Traceable
Pointcut
caller: ITracingParticipant
target: ITraced
m(..)
Traceable.createTrace instantiation
getTrace.Itarget → Icallee
getTrace.ICaller → ICaller
getTrace.Icaller → Icaller
getTrace.InewTrace → tracedOp
getTrace.Im → m
Advice
caller: ITracingParticipant
target ITraced
myContext: TracingContext
myTrace: TraceList
tracedOp := createTrace(Im)
myContext := getContext()
addTrace(tracedOp)
insert(tracedOp)
m(..)
Traceable binding
caller → *
target → *
m → + * Traceable.*(..)
message view wasModified
Pointcut
caller: Caller
target: TracingContext
Iresult := wasModified(ITraced o)
Advice
Iresult
caller: Caller
target: TracingContext
myTrace: TraceList
Iresult := wasModified(ITraced o)
ops := findTraces(o)
loop [t within elements]
t: Trace
include := wasAppliedTo(o)
loop [i within ops, found ≠ true]
i: Trace
found := isModify()
Binding
caller → *
Caller → *
target → *
message view getModified
Pointcut
caller: Caller
target: TracingContext
Iresult[] := getModified()
Advice
Iresult
caller: Caller
target: TracingContext
myTrace: TraceList
Iresult[] := getModified()
Iresult := findModified()
loop [t within elements]
t: Trace
modif := isModify()
opt [modif]
include := getTraceable()
Binding
caller → *
Caller → *
target → *
message view removeTraces
Pointcut
caller: Caller
target: TracingContext
removeTraces(Trace t[])
Advice
caller: Caller
target: TracingContext
myTrace: TraceList
removeTraces(Trace t[])
removeAll(t)
loop [tr within t]
tr: Trace
Binding
caller → *
Caller → *
target → *


<<destroy>>
nessage view addTraces
Pointcut
Advice
caller: Caller
target: TracingContext
addTraces(Trace t[])
caller: Caller
target: TracingContext
myTrace: TraceList
addTraces(Trace t[])
addAll(t)
Binding
caller → *
Caller → *
target → *