

aspect EntrySynchronizing depends on Collaborative, Blockable

structural view

IEntrySynchronizingParticipant

EntrySynchronizingContext

- int minParticipantCount  
- int waitingParticipantCount  
- int getMinParticipantCount()  
+ setMinParticipantCount(int)  
- int getWaitingParticipantCount()  
~ increaseWaitingParticipantCount()  
~ decreaseWaitingParticipantCount()  
~ Semaphore getEntrySemaphore()  
~ releaseEntrySemaphore()  
~ boolean isLastEnteringParticipant

IEntrySynchronizingParticipant

+ createAndEnterContext()  
+ joinContext(EntrySynchronizingContext)  
- waitForEntrySemaphore()

Collaborative instantiation

ICollaborativeParticipant →  
IEntrySynchronizingParticipant

Blockable instantiation

IBlockableParticipant →  
IEntrySynchronizingParticipant

Blockable binding

EntrySynchronizingContext → BlockableContext  
entrySemaphore → semaphore  
getEntrySemaphore → getSemaphore  
waitForEntrySemaphore → waitForSemaphore  
releaseEntrySemaphore → releaseSemaphore

Collaborative binding

EntrySynchronizingContext →  
CollaborativeContext  
createAndEnterContext → createAndEnterContext  
joinContext → joinContext

message view enterContext

Pointcut

caller: Caller

target:  
IEntrySynchronizingParticipant

enterContext(c)

Advice

caller: Caller

target:  
IEntrySynchronizingParticipant

enterContext()

\*

waitOrReleaseBeforeEntering(c)

Default

Instantiation

caller → \*

Caller → \*

target → \*

message view waitOrReleaseBeforeEntering affected by Collaborative.getContext, Blockable.releaseSemaphore, Blockable.waitForSemaphore

Pointcut

caller: Caller

target:  
IEntrySynchronizingParticipant

waitOrReleaseBeforeEntering(c)

Advice

caller: Caller

target:  
IEntrySynchronizingParticipant

waitOrReleaseBeforeEntering(c)

context := getContext()

opt [context = c]

increaseWaitingParticipantCount()

last := isLastEnteringParticipant()

alt

[last]

releaseEntrySemaphore()

[else]

waitForEntrySemaphore()

decreaseWaitingParticipantCount()

c:  
EntrySynchronizingContext

Default

Instantiation

caller → \*

Caller → \*

target → \*

message view isLastEnteringParticipant

Pointcut

caller: Caller

target:  
EntrySynchronizing  
Context

last := isLastEnteringParticipant()

caller: Caller

Advice

target:  
EntrySynchronizing  
Context

last := isLastEnteringParticipant()

last := false

waiting := getWaitingParticipantCount()

min := getMinParticipantCount()

opt [waiting >= min]

last := true

Default

Instantiation

caller → \*

Caller → \*

target → \*