SWEN 223

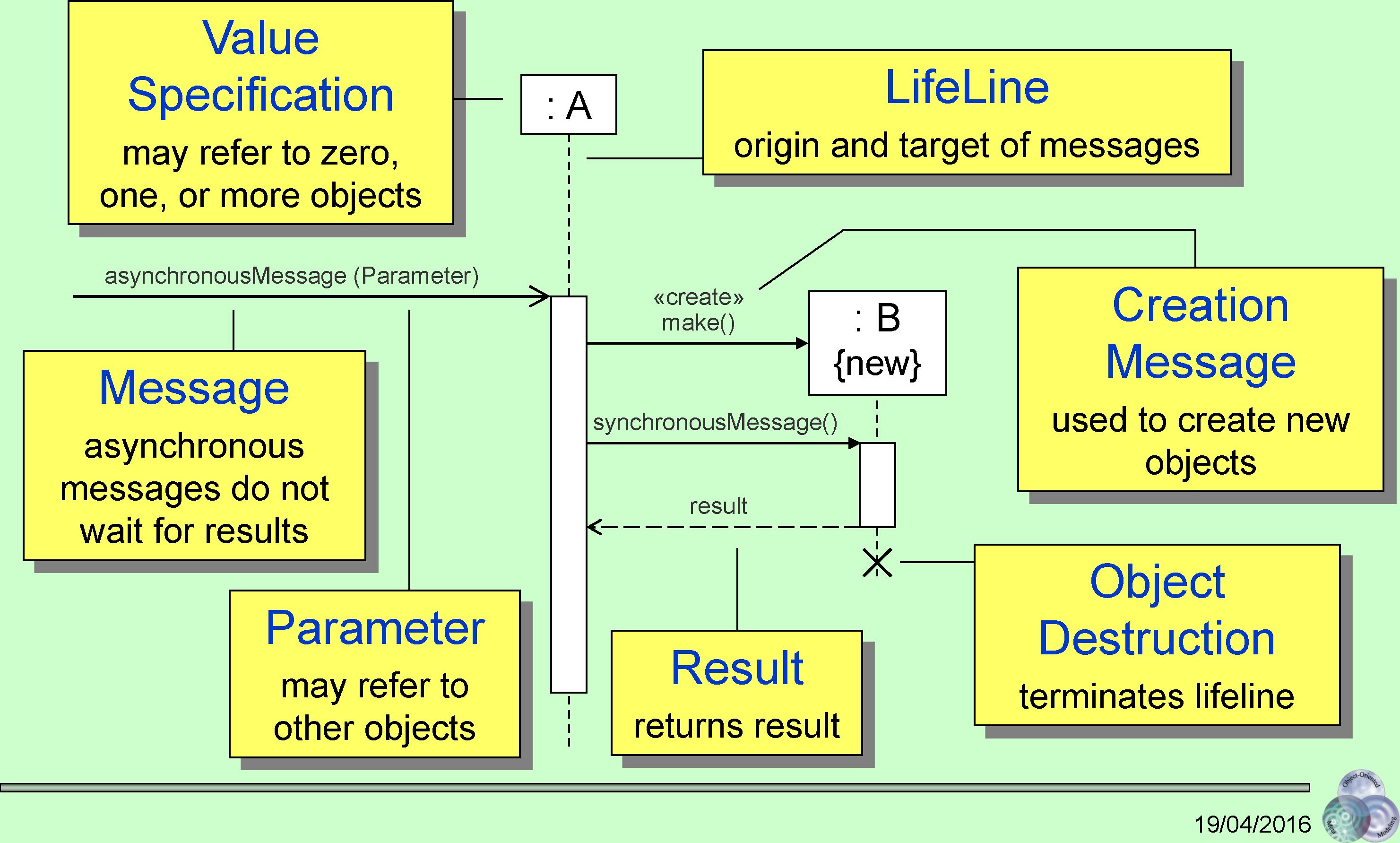
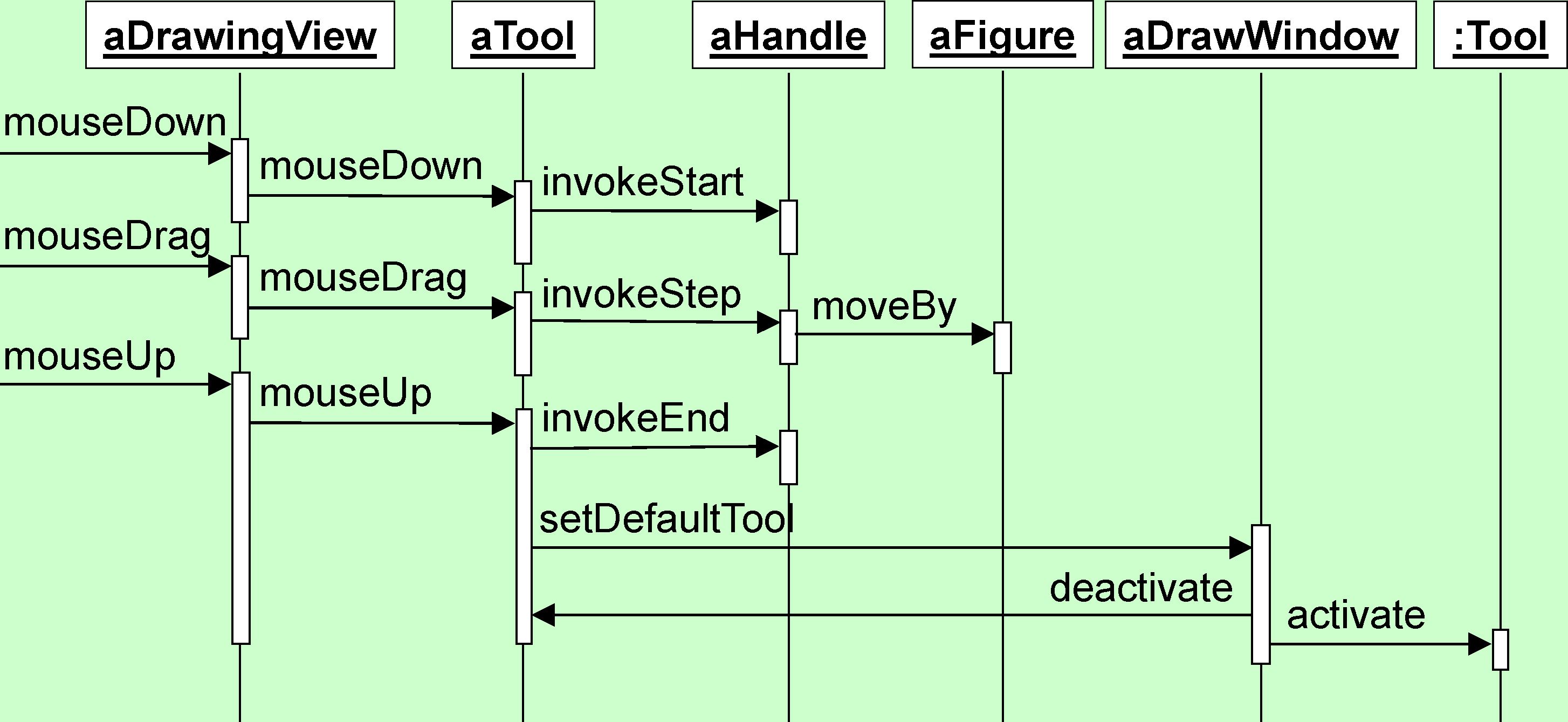
Software Engineering Analysis  
Interaction Diagrams

Thomas Kuhne

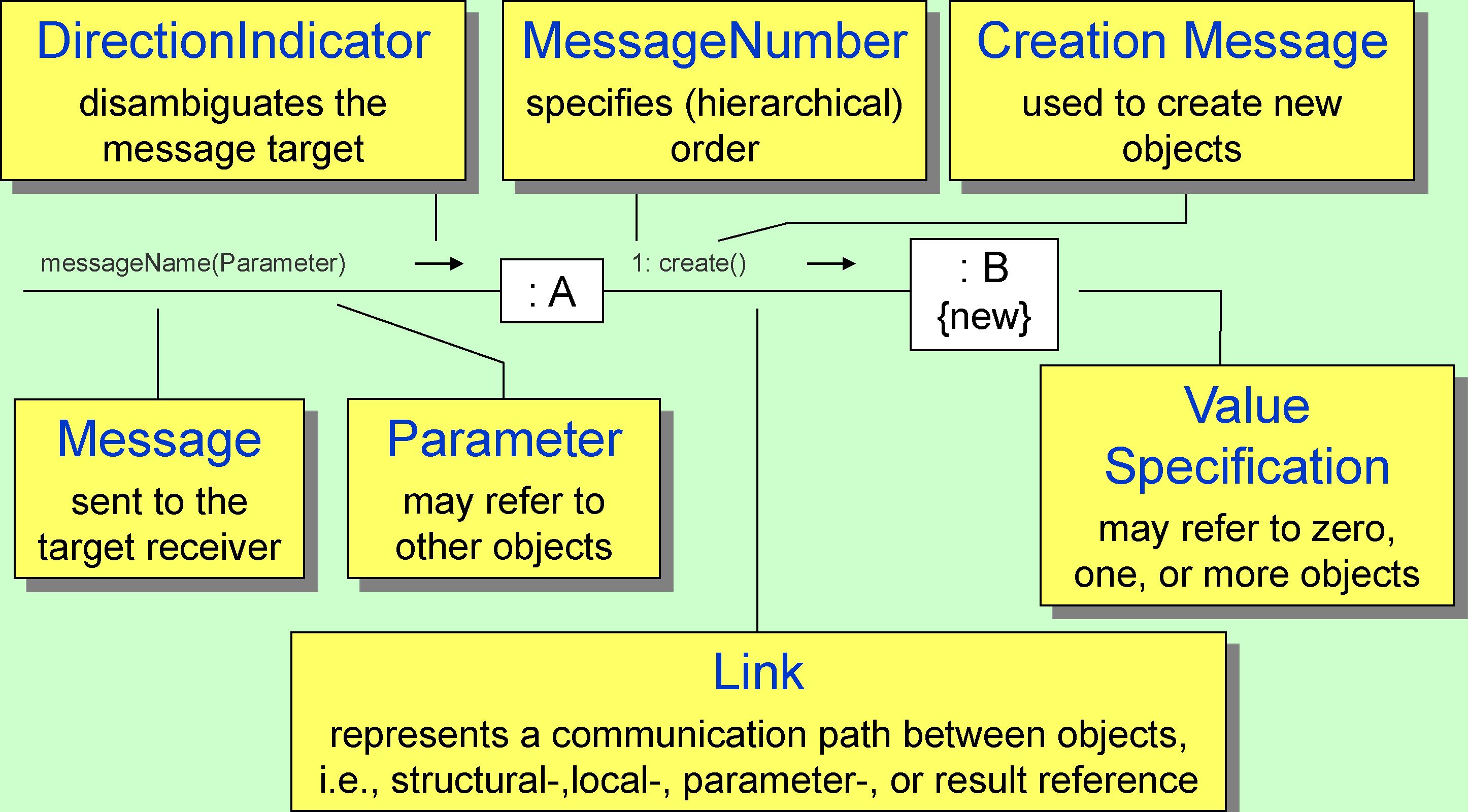
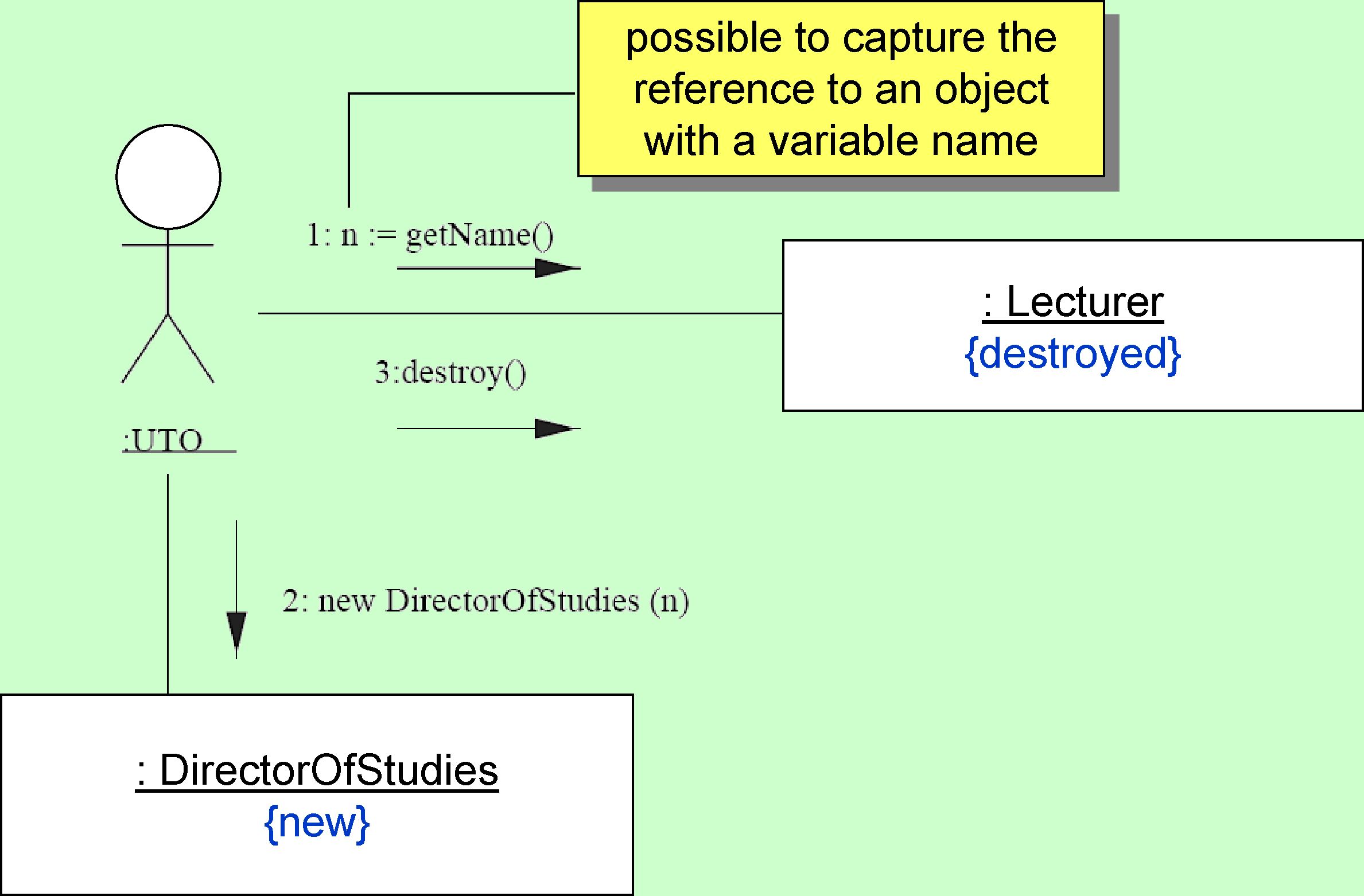
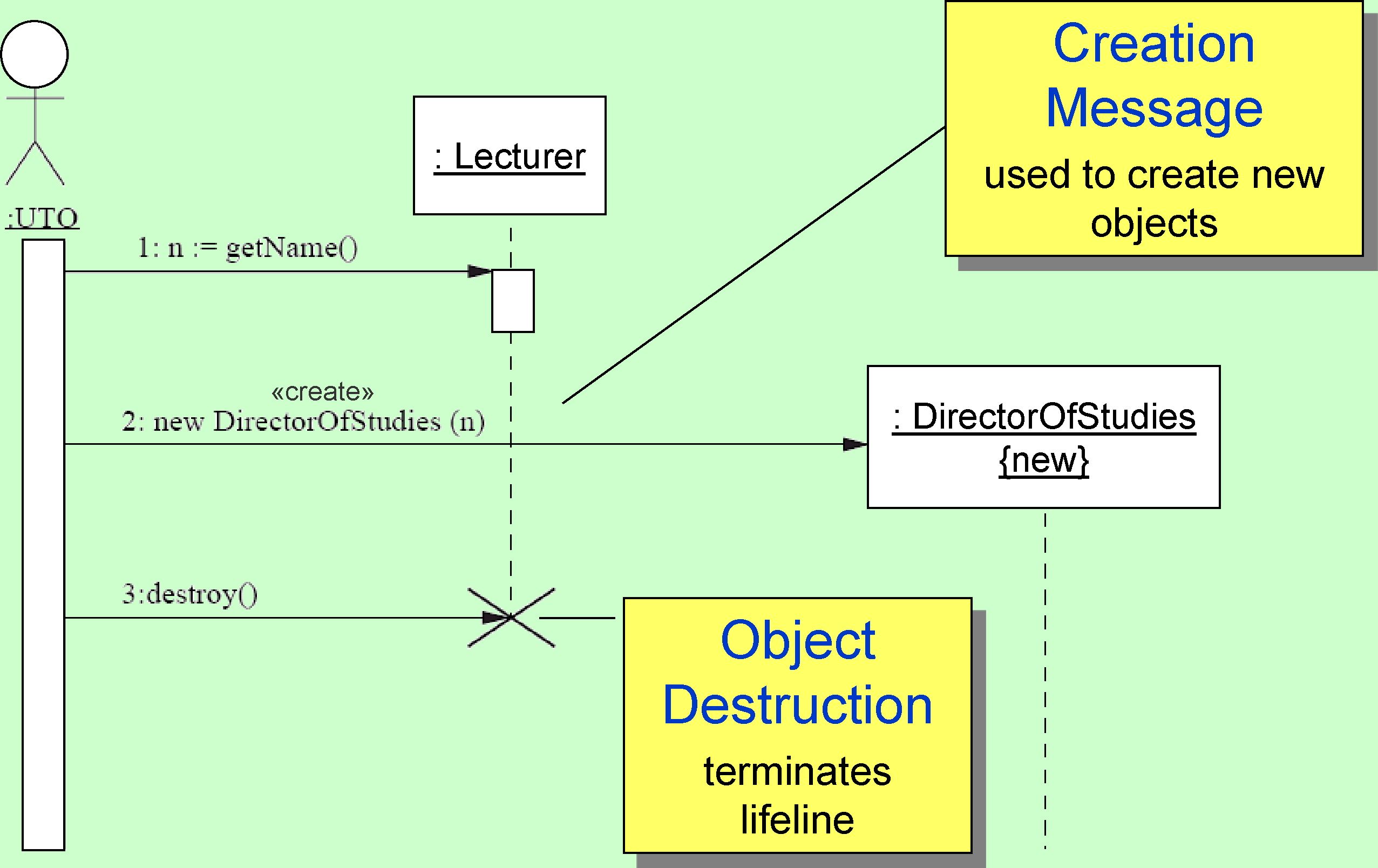
Victoria University of Wellington  
[Thomas.Kuehne@ecs.vuw.ac.nz](mailto:Thomas.Kuehne@ecs.vuw.ac.nz), Ext. 5443, Room Cotton 233



sqpi Documenting Behaviour



Creation & Deletion



-I ,ru\.

Annotations

• Possible Object/Link Existence Classification

» new

* the instance or link is created during the enclosing interaction

» destroyed

* the instance or link is destroyed prior to the completion of the enclosing interaction

» transient

* the instance or link is created during execution but is destroyed before completion of the enclosing interaction



Specifying Behaviour

* Message Guards

» [ pressure > 9 ]: playAlarm()

» the message is sent only if the condition evaluates to true

» are deprecated for sequence diagrams; use “opt” or “alt” notation instead

* Iteration

iteration

marker

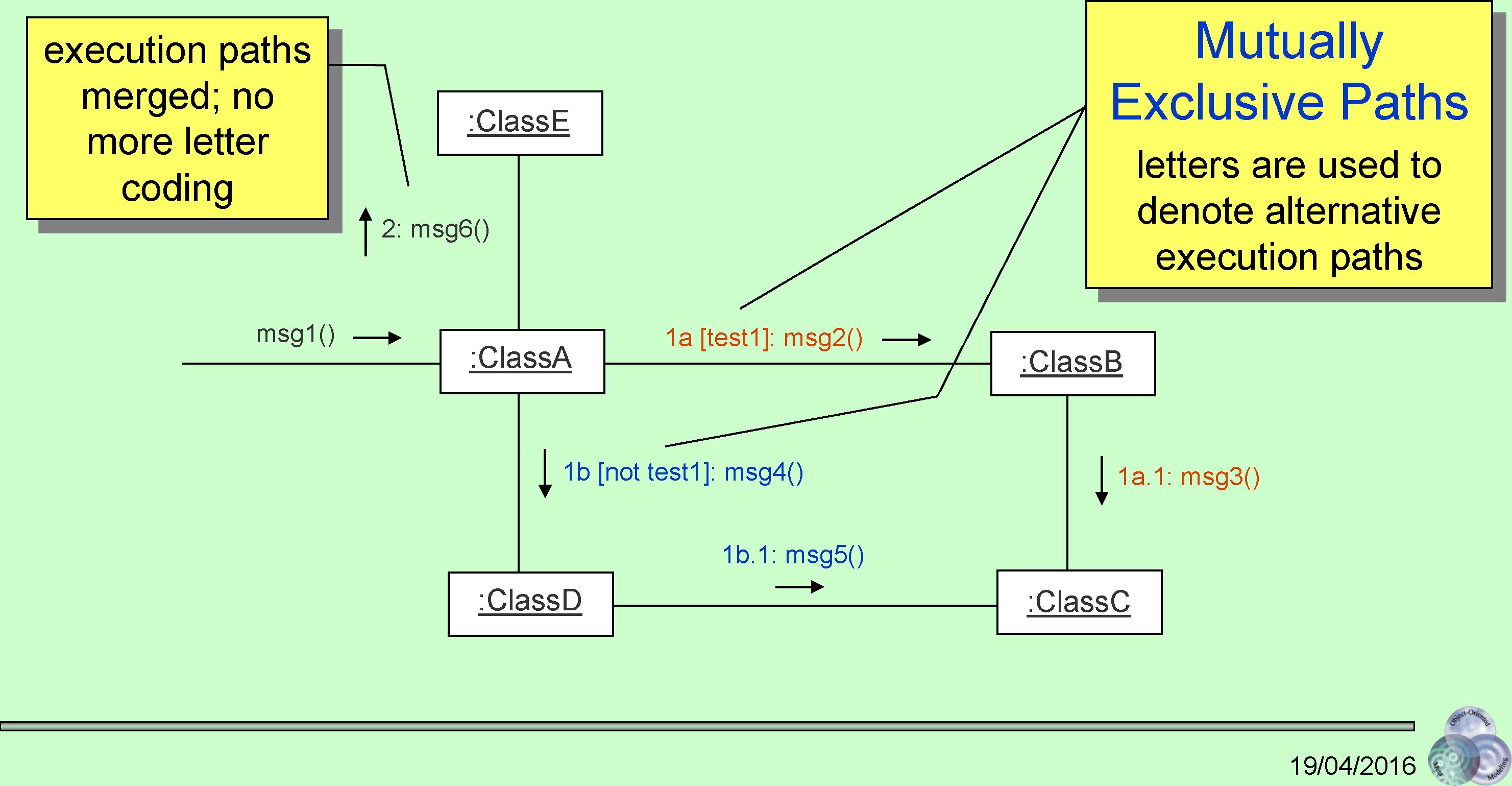
» \* [ i := 1..n ]: knockAtDoor() /

» conditions such as \* [ x<10 ] or \* [ isNotEmpty ] are possible as well



Conditional Paths

Extension of Hierarchical Notation



Strength

Sequence Diagrams

:ClassBInstance

message1()

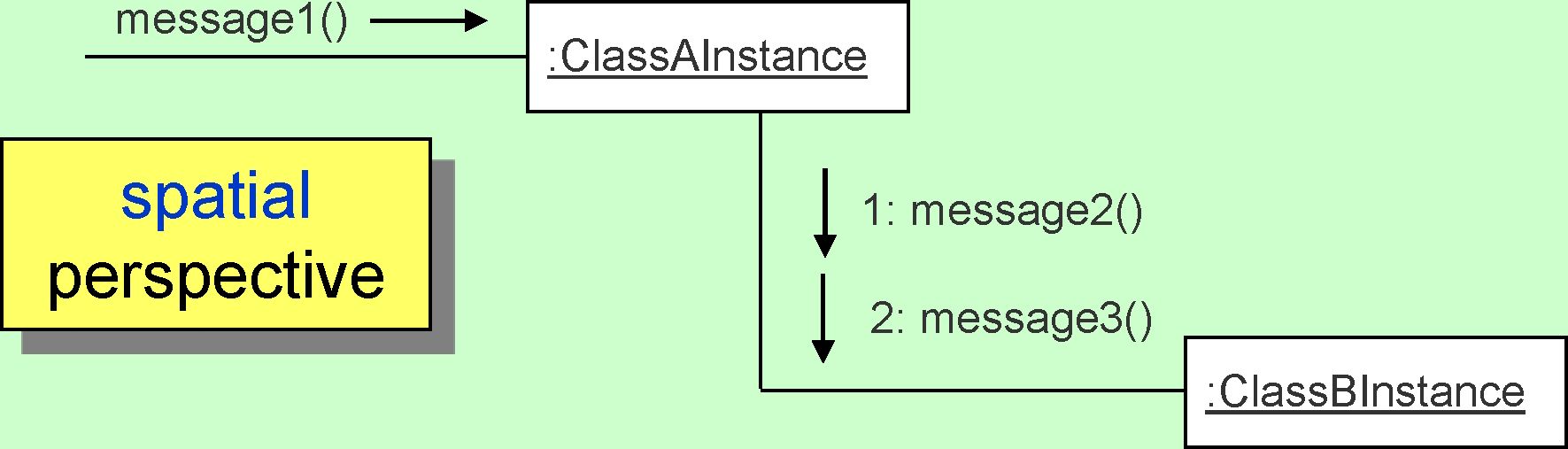
:ClassAInstance

1

|  |  |  |
| --- | --- | --- |
| 2 U » |  | i  i  message2() \ |
| temporal 1 |  | message3() i |
| perspective |  | 1  1 |

i i

Communication Diagrams



clearly show ordering of messages

Weakness

don’t show links become very wide

Strength

show links & use space economically

Weakness

difficult to see message sequence



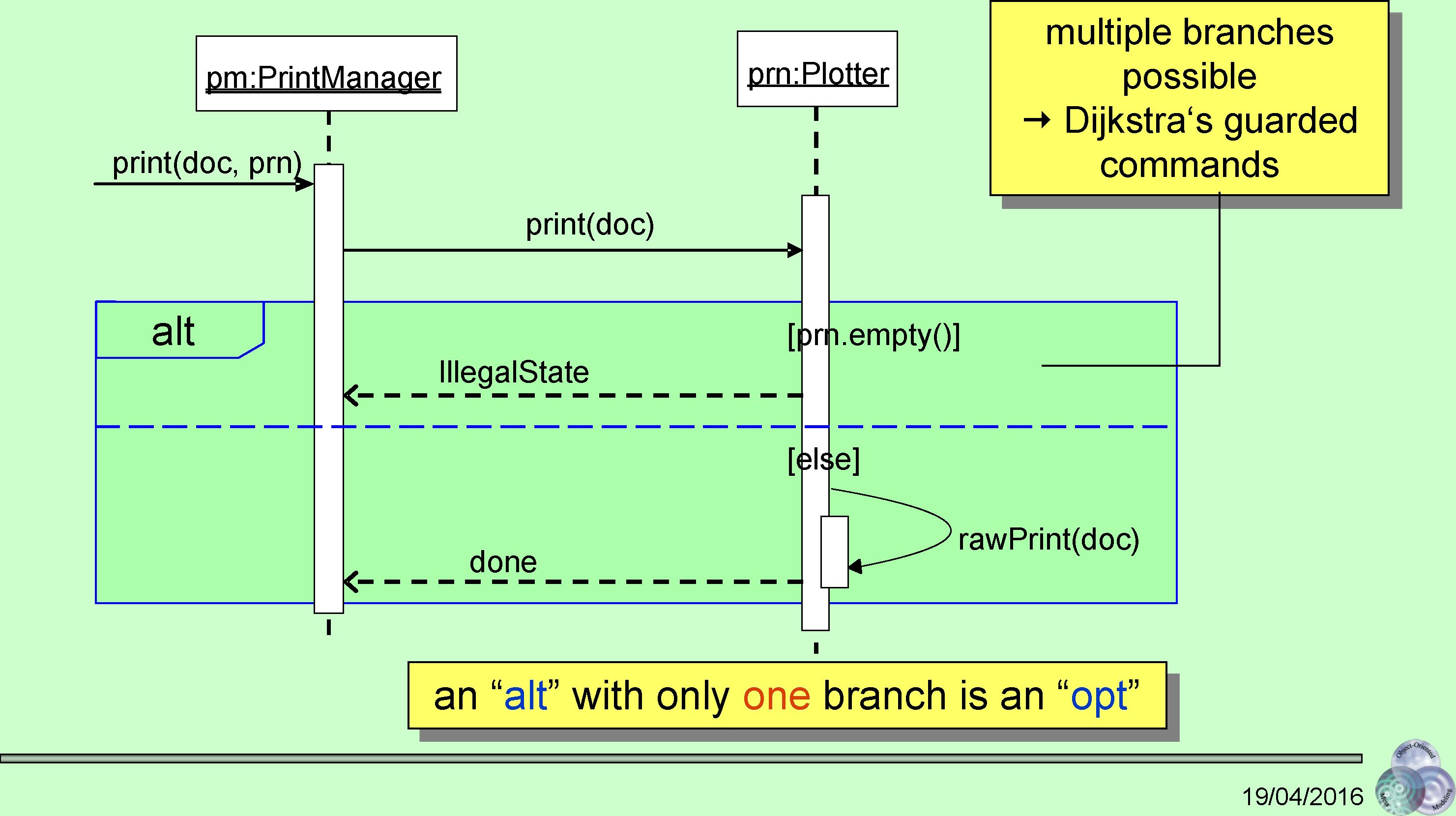
* As of UML 2.0, Sequence Diagrams have a lot more expressive notation than Communication Diagrams

» alternatives, loops » decomposition mechanisms

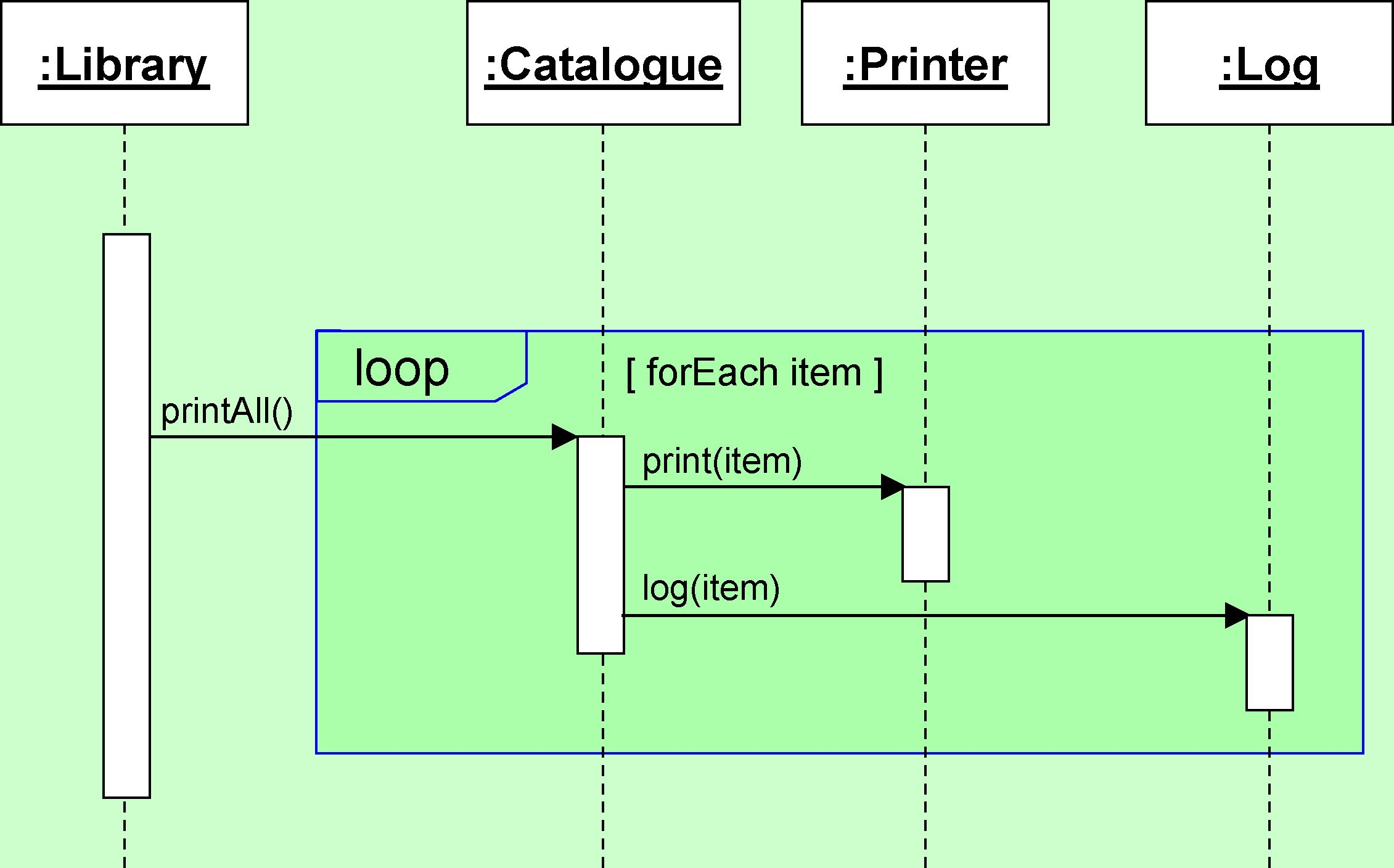
* With respect to their common basis, both diagram kinds can be translated into each other



Conditional



Iteration



sqpi Diagram Decomposition

Interaction

sd GoHomeSetup

/

|  |  |  |  |
| --- | --- | --- | --- |
| :ServiceUser |  | :ServiceBase |  |

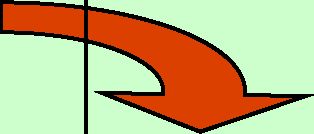
:ServiceTerminal

Occurrence

include interactions defined elsewhere

ref

Authorization



sd Authorization

|  |  |  |
| --- | --- | --- |
| \ 1 1 1  opt) [ serviceUnknown] | | |
| ref J | FindLocation | |
|  | setHome  ^1 |  |
|  |
|  | setInvocationTime |  |
| setTransportPreferences i |
|  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| :ServiceUser |  | :ServiceBase |  | :ServiceTerminal |
|  |  |  |  |  |

code

OK

K-

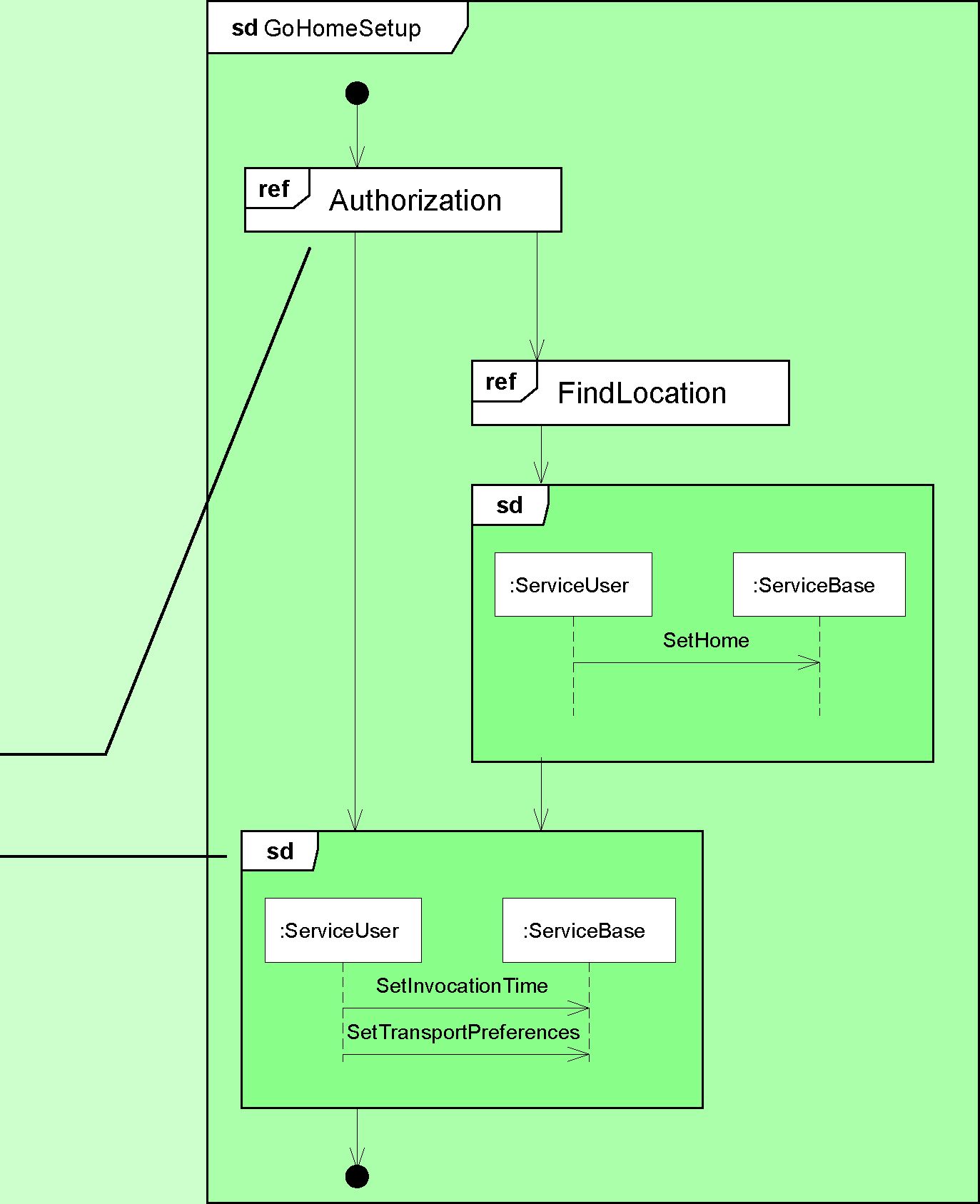
onWeb

OK



Interaction Overview Diagram

• Variation of Activity Diagrams



» provide overview of the control flow

» combines sequence fragments

* interaction occurrences
* interactions

Interaction Diagrams

Applicability

* during analysis, to improve individual or group understanding of inter-object behaviour

» are all communication paths required available?

» can complete message sequences be constructed?

» documentation for CRC scenarios

* during design, to precisely (but typically partially) describe inter-object/process communication
* during testing, the traces can be compared with those described in the earlier phases

