# **C**onception **O**rientée **O**bjets
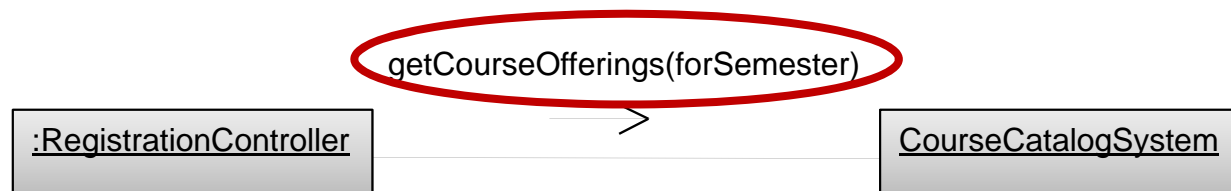
## **Interactions**

Frédéric Mallet

http://deptinfo.unice.fr/~fmallet/

# Interactions

❑ Objects are useless unless they can collaborate to solve a problem.

  ▪ Each object is responsible for its own behavior and status.

  ▪ No object can carry out every responsibility on its own.

❑ How do objects interact with each other?

  ▪ They interact through messages.

  ▪ A message shows how one object asks another object to perform some activity.

getCourseOfferings(forSemester)

| :RegistrationController | | CourseCatalogSystem |

# What is an Interaction Diagram?

❑ Generic term that applies to several diagrams that emphasize object interactions

- Sequence Diagram
- Communication Diagram

❑ Specialized Variants

- Timing Diagram
- Interaction Overview Diagram

❑ In UML 2.0, heavily inspired from

- Message Sequence Charts (ITU-T SDL, Z.120)
- Live Sequence Charts (D. Harel)
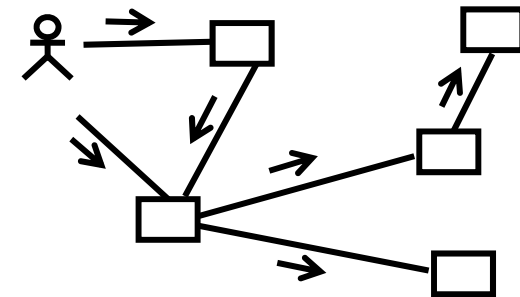
# Interaction Diagrams

## ❑ Sequence Diagram

- ■ Time oriented view of object interaction
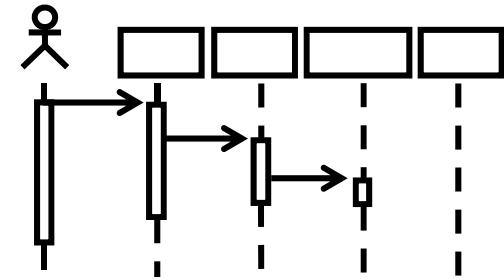
Sequence Diagrams

## ❑ Communication Diagram

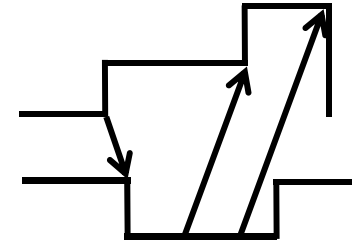- ■ Structural view of messaging objects

Communication Diagrams
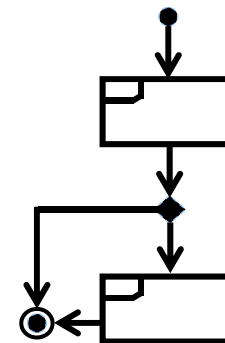
# Interaction Diagrams

❑ **Timing Diagram**

- ■ Time constraint view of messages involved in an interaction



Timing Diagrams

❑ **Interaction Overview Diagram**

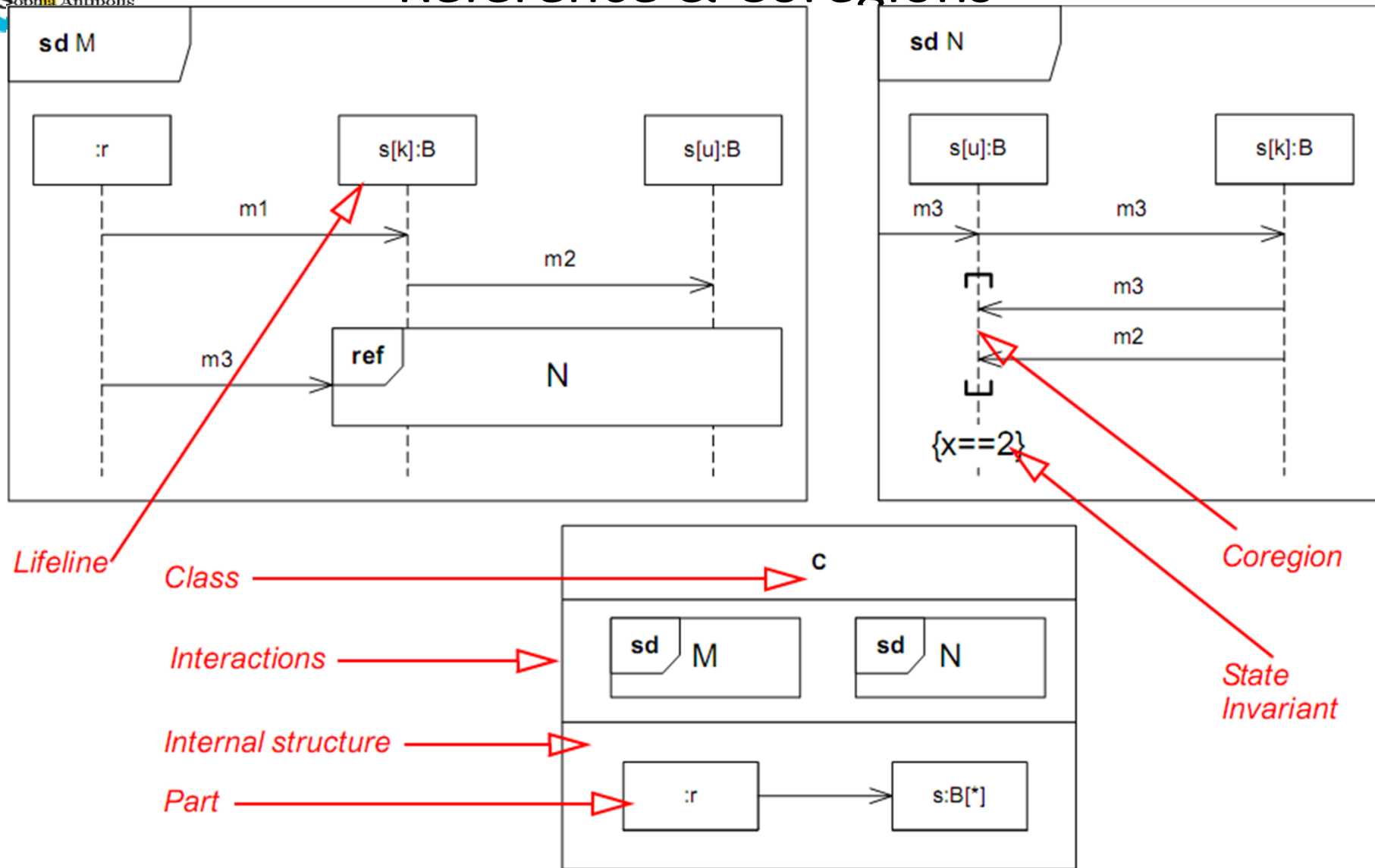- ■ High level view of interaction sets combined into logic sequence
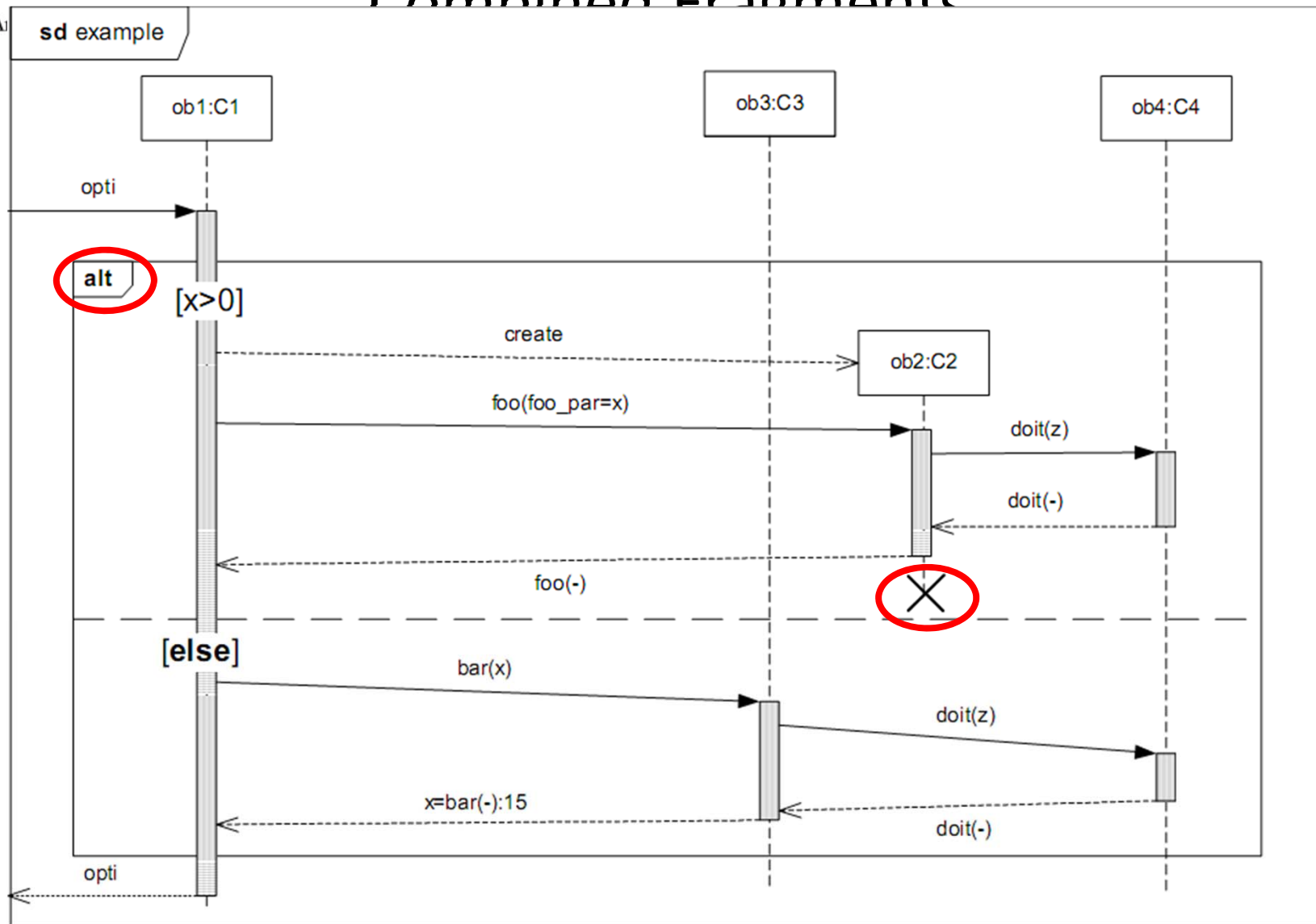


Interaction Overview Diagrams

# Classes & Interactions

❑ Les interactions peuvent décrire

- Un comportement interne d'une classe
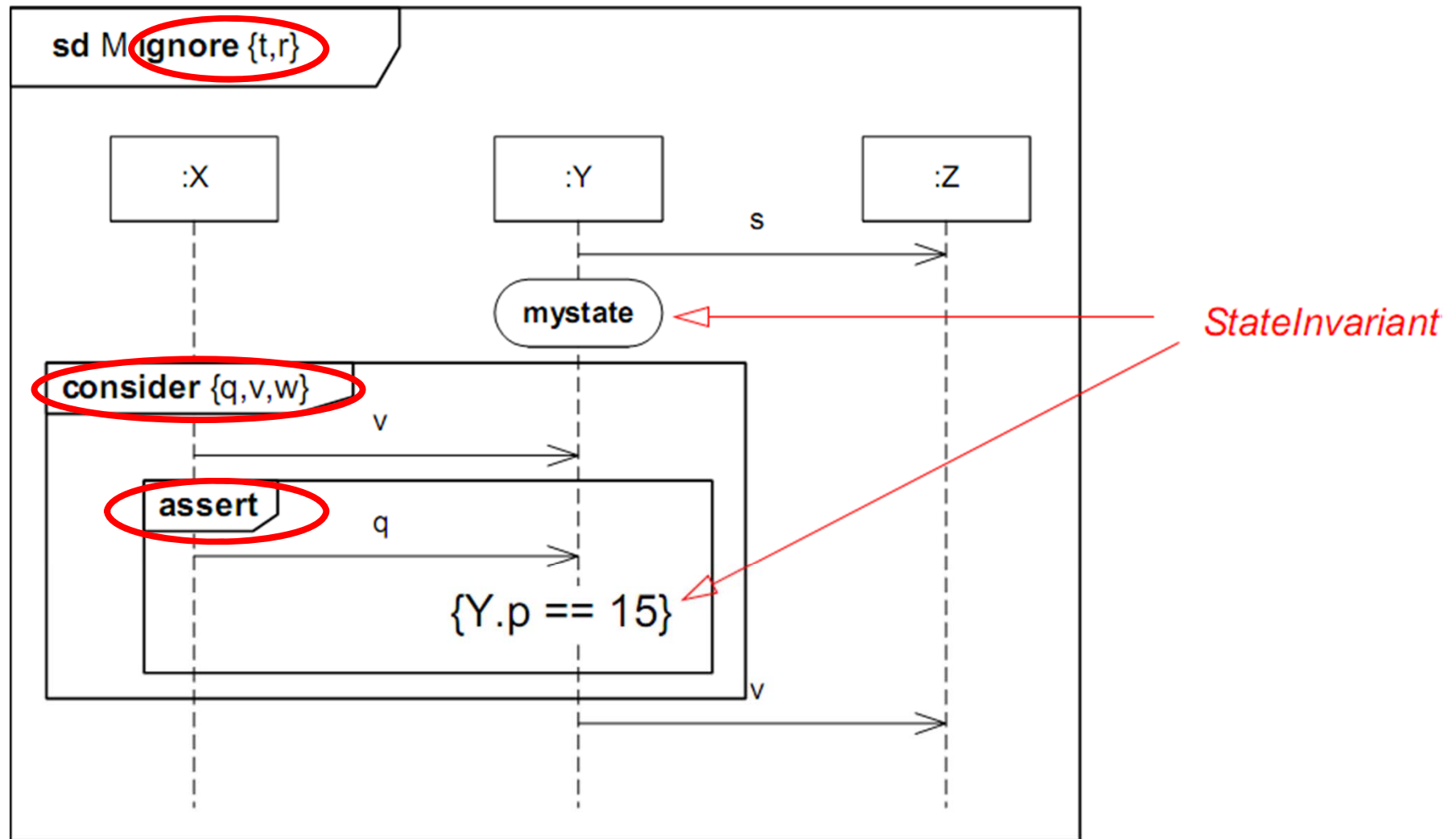- Les interactions entre les "parts" d'une classe structurée ou d'une collaboration

# Reference & Coregions

# Combined Fragments

# Consider, Assert, Ignore

# Primary Interaction Operators

- ❑ **ref** name
  - ▪ reference to a sequence diagram fragment defined elsewhere
- ❑ **opt** [condition]
  - ▪ has 1 part that may be executed based on a condition/state value
- ❑ **alt**
  - ▪ has 2 or more parts, but only one executes based on a condition/state
  - ▪ an operand fragment labeled [else] is executed if no other condition is true
- ❑ **par**
  - ▪ has 2 or more parts that execute concurrently
  - ▪ If there is only one processor the behavior could be (A then B), (B then A), or (A and B interleaving) …
- ❑ **loop** min..max [escape]
  - ▪ Has a minimum # of executions, and optional maximum # of executions, and optional escape condition
- ❑ **break** [condition]
  - ▪ Has an optional guard. If true, the contents (if any) are executed, and the remainder of the enclosing operator is not executed

**Provided by Michael Chonoles**

# Other Interaction Operators

❑ **critical**

- The sequence diagram fragment is a critical region. It is treated as atomic – no interleaving with parallel regions

❑ **neg**

- The sequence diagram fragment is forbidden. Either it is impossible to occur, or it is the intent of the requirements to prevent it from occurring

❑ **assert**

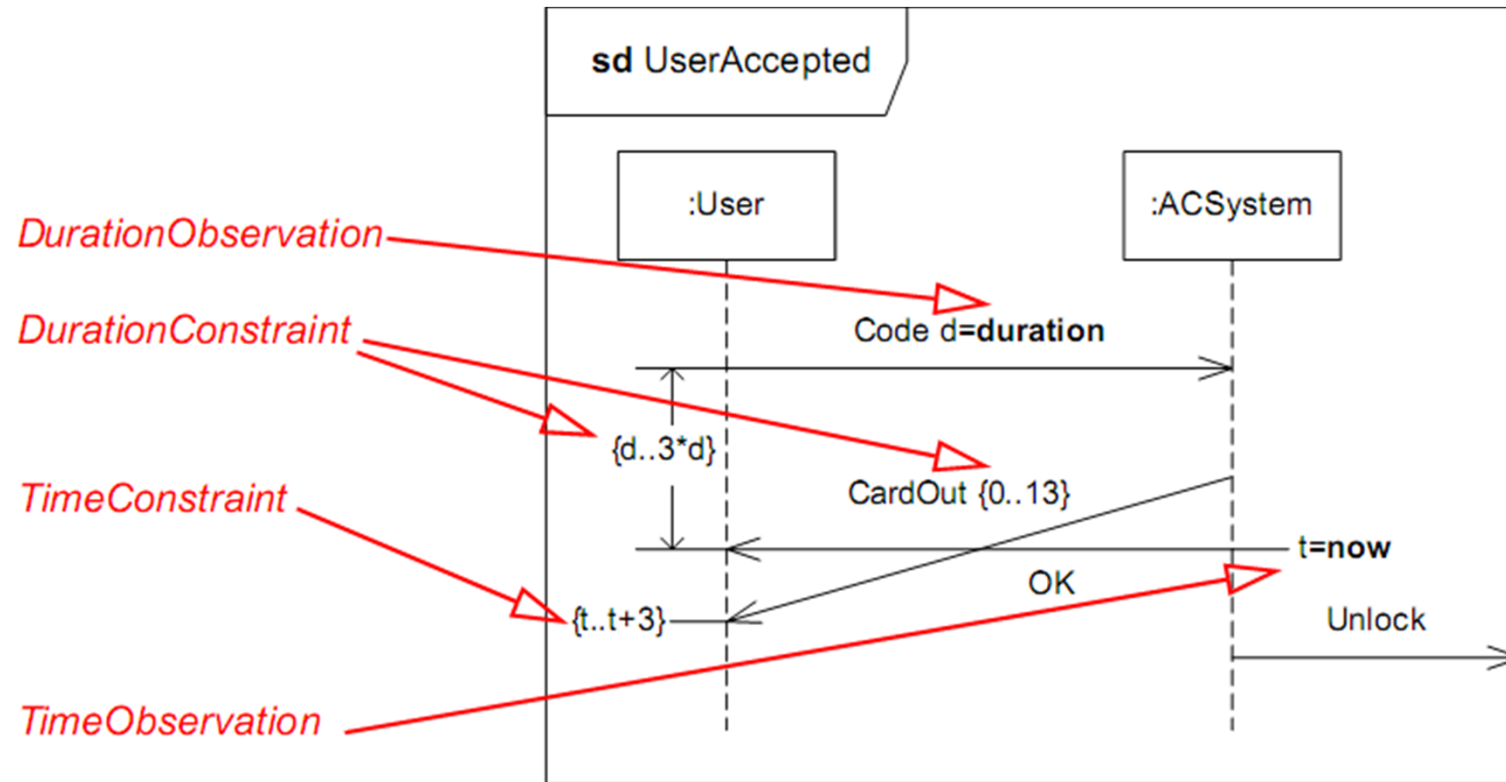- The sequence diagram fragment is the only one possible (or legal)

❑ **seq**  (weak, the default) / **strict**

- Strict: The message exchange occurs in the order described
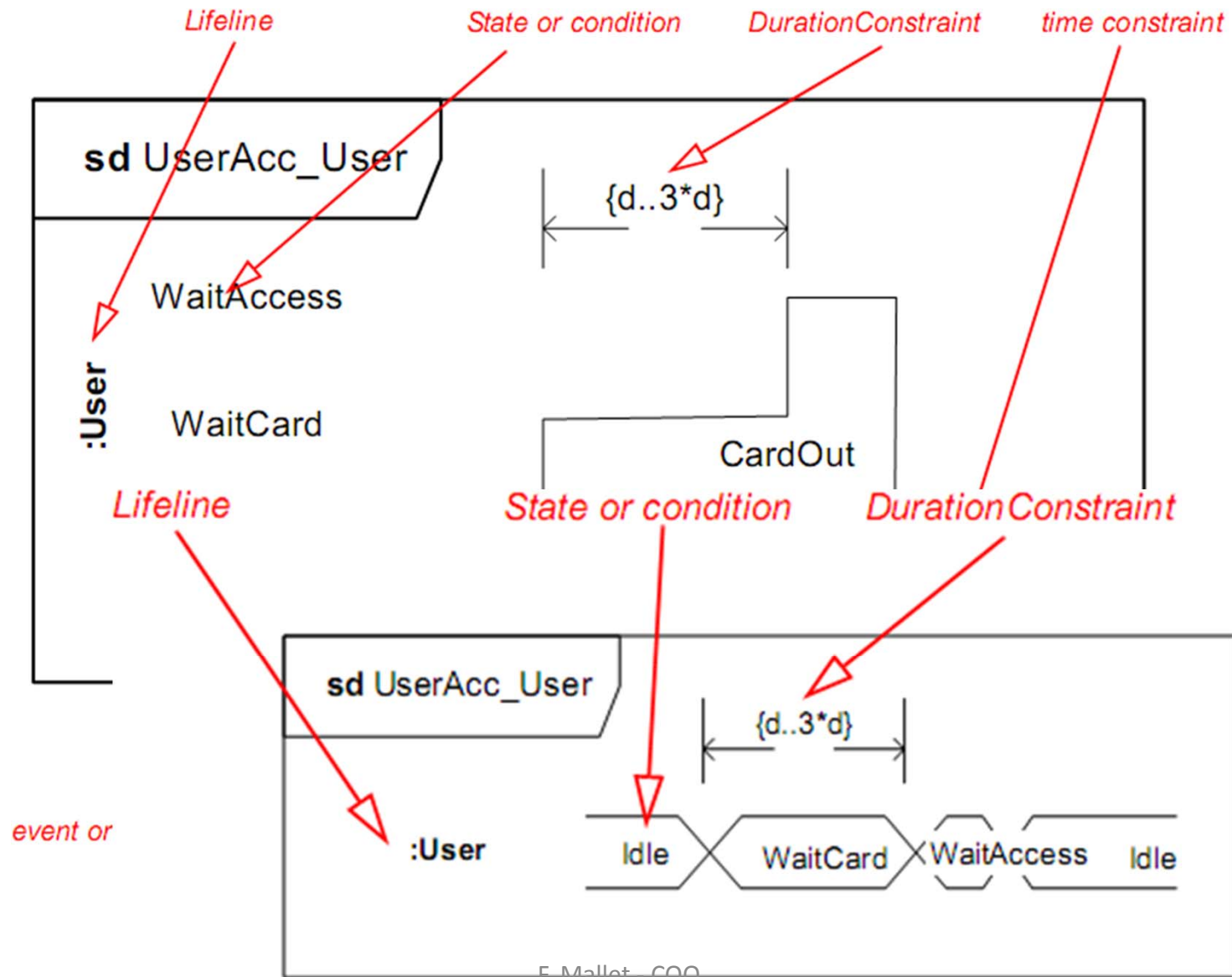- Weak: Each lifeline may see different orders for the exchange

❑ **consider** (list of messages) / ig**nore** (list of messages)

- Consider: List the messages that are relevant in this sequence fragment
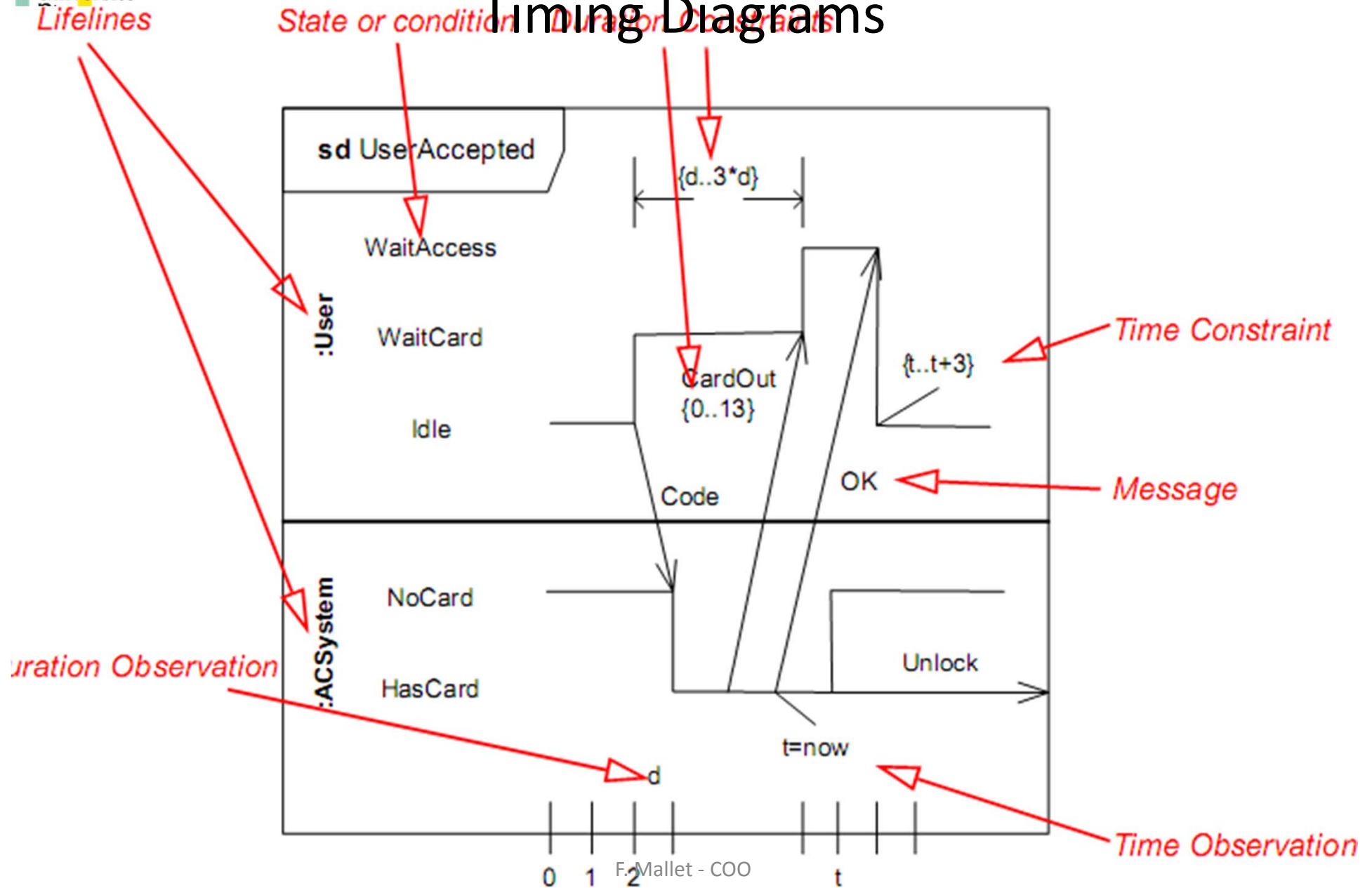- Ignored:   List the messages that may arrive, but are not interesting here

**Provided by Michael Chonoles**

# Observations & Constraints

# Timing Diagrams

# Timing Diagrams

*Lifelines*

*State or condition timing or Constraint*

*Time Constraint*

*Message*

*uration Observation*

*Time Observation*

**sd** UserAccepted

{d..3*d}

WaitAccess

:User

WaitCard

Idle

CardOut
{0..13}

Code

{t..t+3}

OK

:ACSystem

NoCard

HasCard

Unlock

d

t=now

0  1  2                    t

# Interaction Overview Diagrams