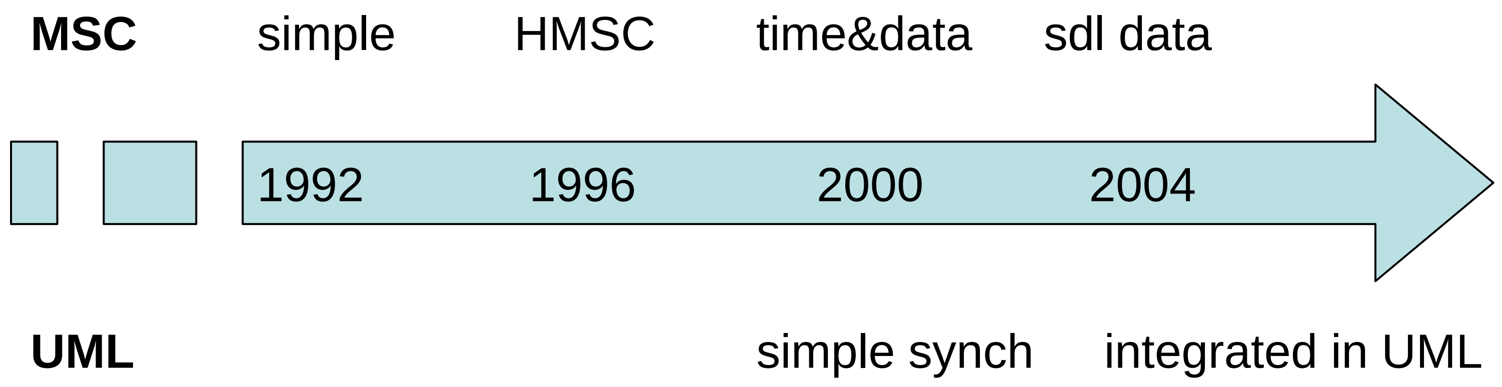
UNIVERSITY
OF OSLO

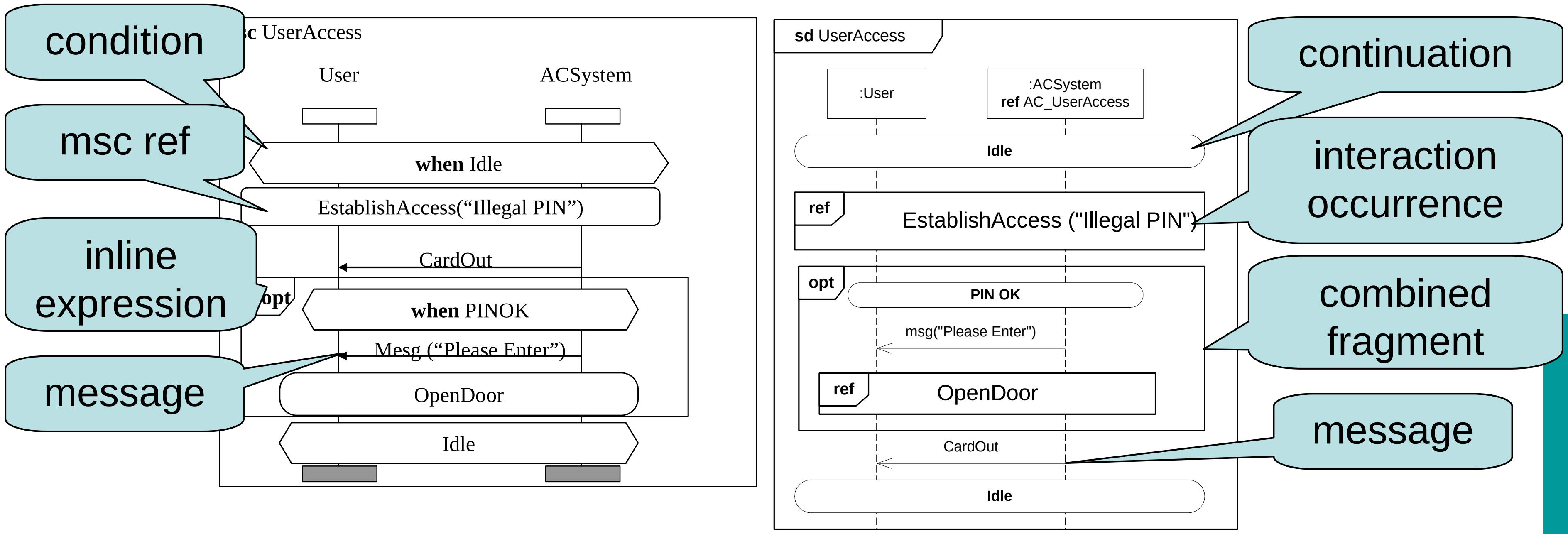
Comparing UML 2.0 Interactions and MSC-2000

Can MSC be retired?

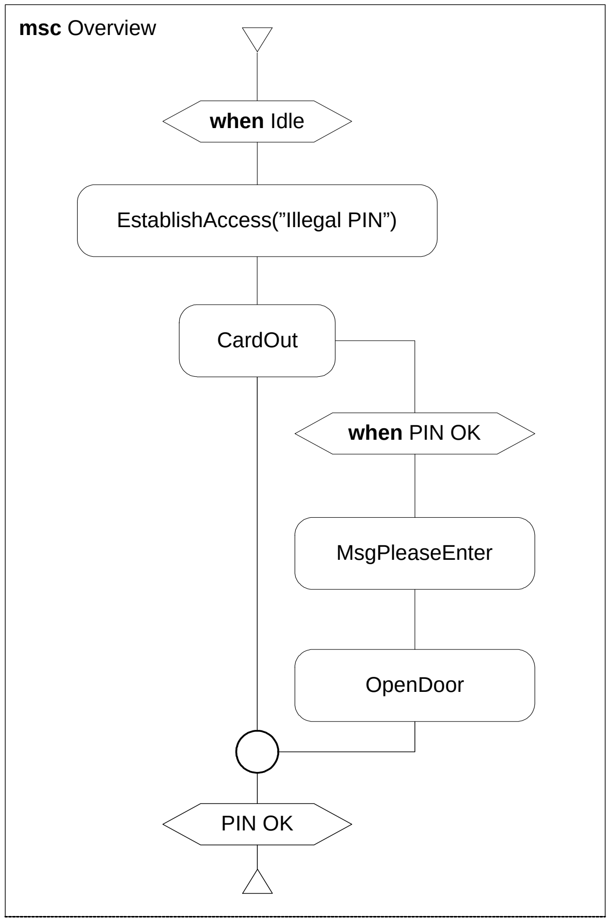
Timeline

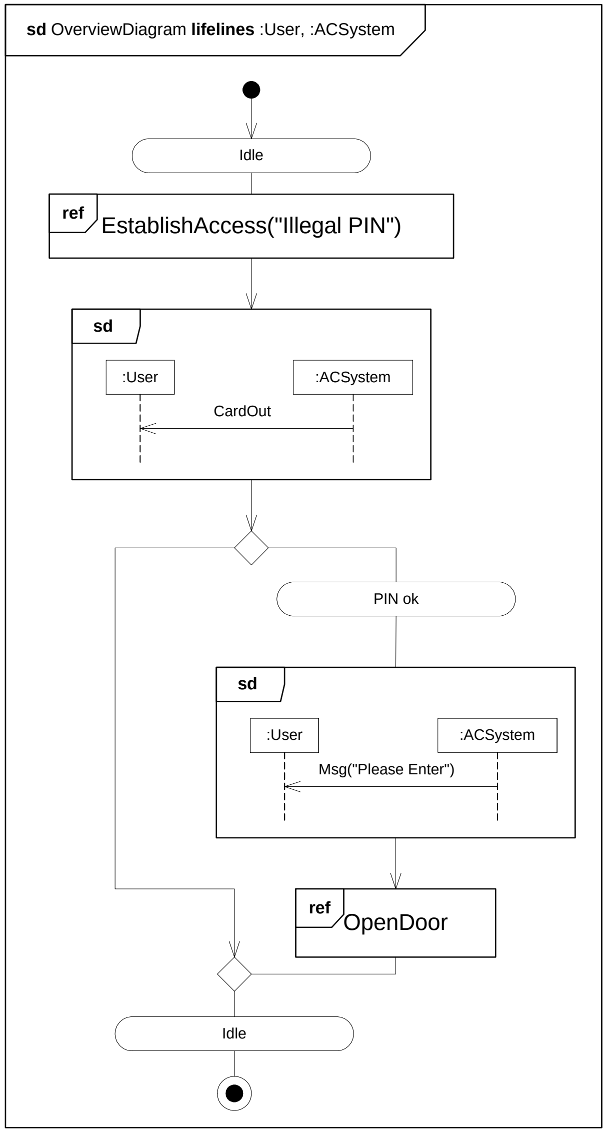

MSC
simple
HMSC
time&data
sdl data
1992
1996
2000
2004
UML
simple synch
integrated in UML

Different terms – but the same concepts

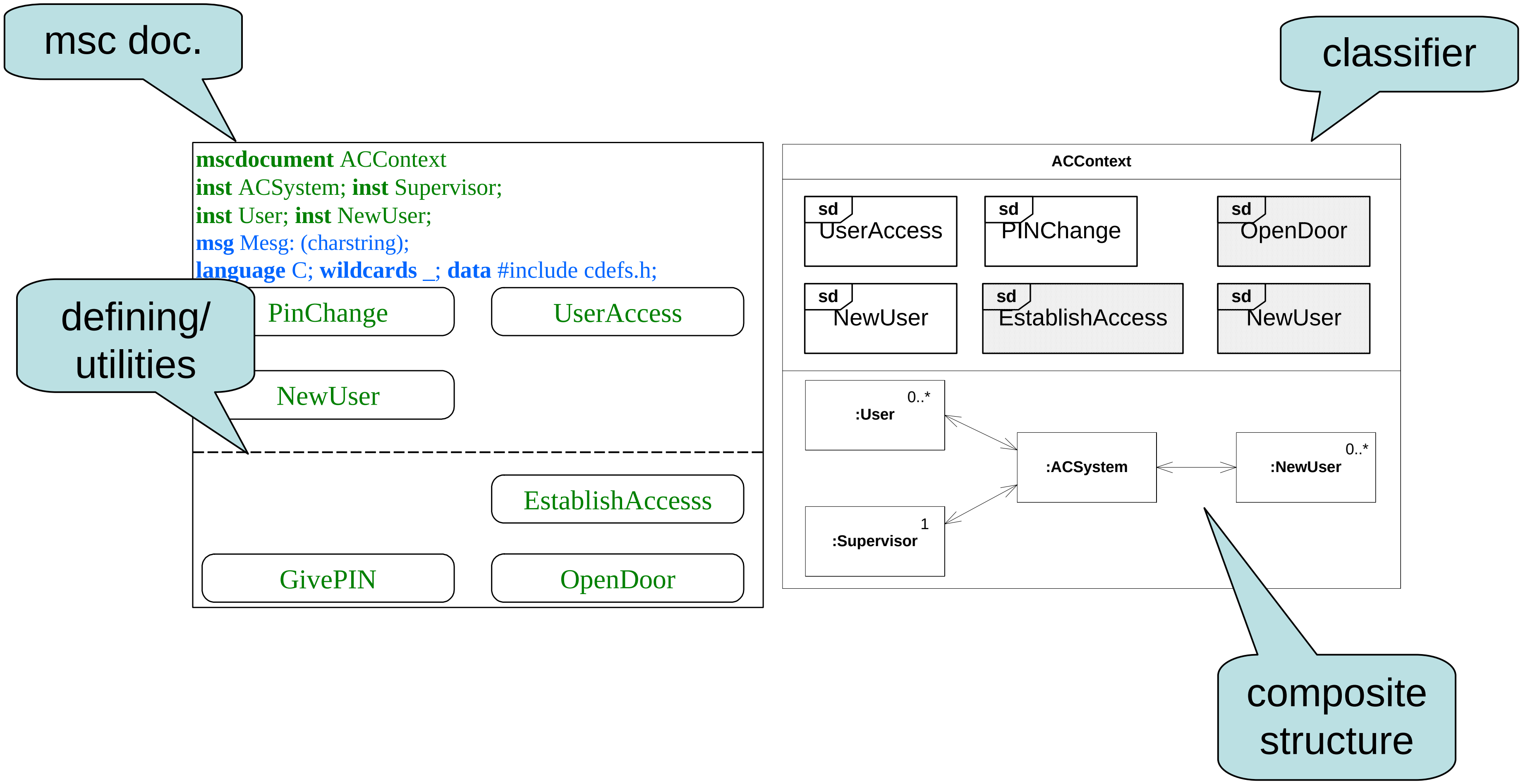


Several kinds of diagrams





Context of MSCs / Interactions



mscdocument ACContext
inst ACSystem; **inst** Supervisor;
inst User; **inst** NewUser;
msg Mesg: (charstring);
language C; **wildcards** _; **data** #include cdefs.h;

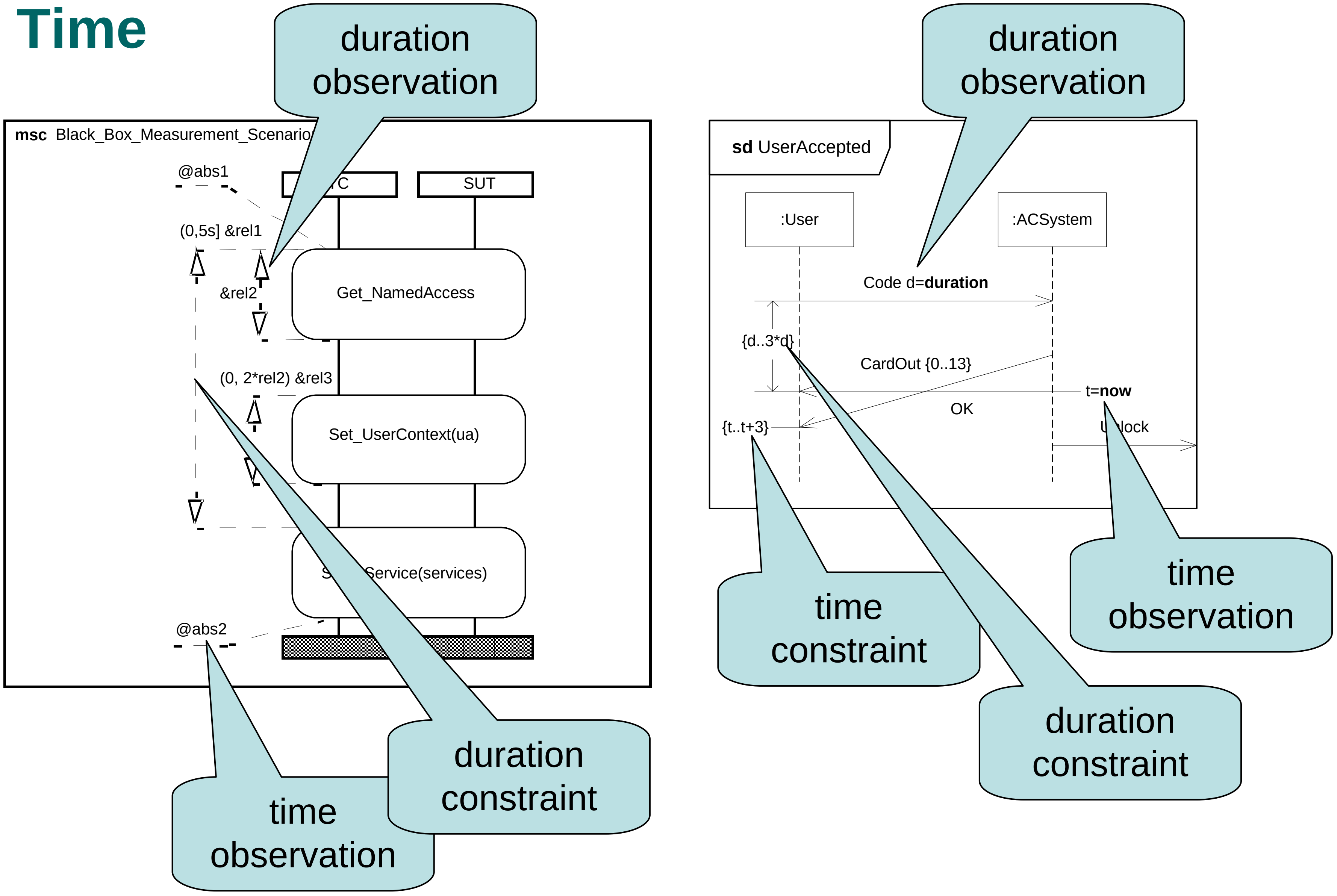
Decomposition, Messages and Suspension region

- Decomposition
  - MSC 2000: hierarchy of MSC Documents
  - UML 2.0: hierarchy of UML classes. Strictly follows the composite structure
- Messages
  - MSC 2000: in general asynchronous, but synchronized method calls can be expressed
  - UML 2.0: synchronous as well as asynchronous. Messages often represent synchronized method calls.
- Suspension region
  - MSC 2000: it is there
  - UML 2.0: it is not there

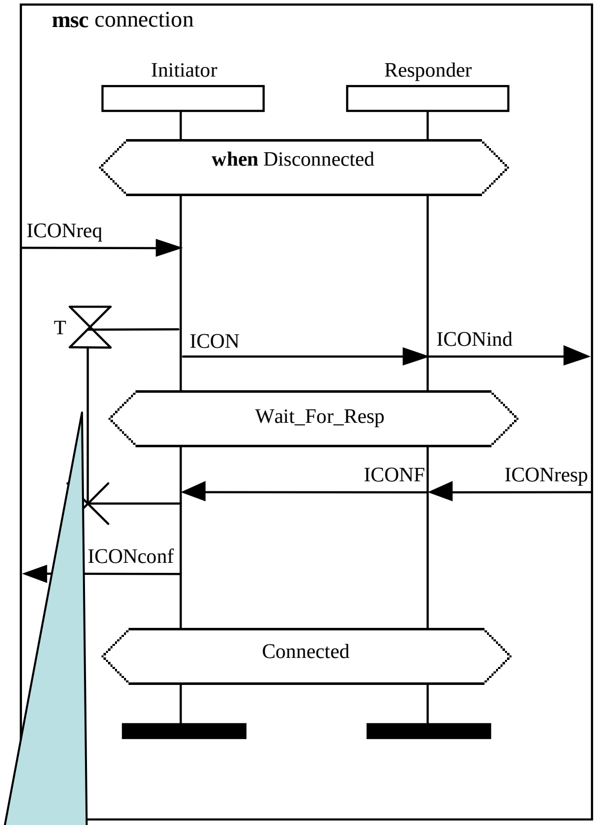
Data

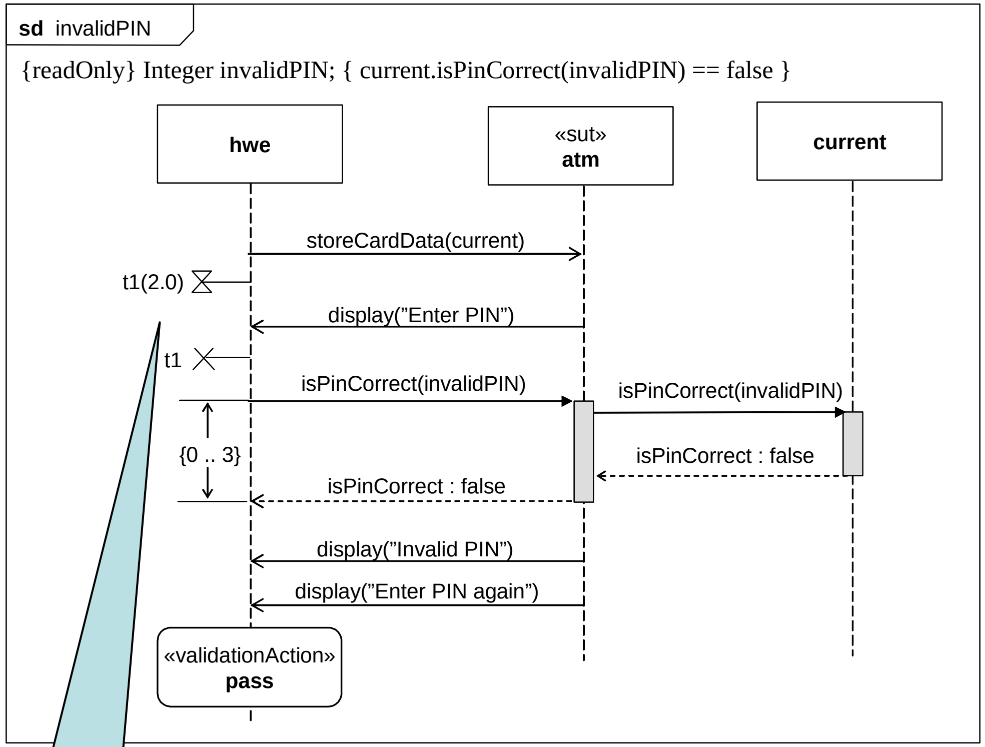
- MSC-2000
  - has a very sophisticated scheme
    - to define the data interface within the description itself
      - both syntax and semantics
    - that will make it possible to use the data language of your choice
  - interface definition exists for SDL (Z.121)
    - does anyone use it?
- UML 2.0
  - has no concrete data language of its own
  - has an abstract syntax of Actions (metamodel)
- In practice
  - both use fragments of programming languages

Time


duration observation
msc Black_Box_Measurement_Scenario
@abs1
C
SUT
(0,5s] &rel1
&rel2
Get_NamedAccess
(0, 2*rel2) &rel3
Set_UserContext(ua)
S Service(services)
@abs2
time observation
duration constraint
duration observation
sd UserAccepted
:User
:ACSystem
Code d=duration
{d..3*d}
CardOut {0..13}
t=now
OK
{t..t+3}
Unlock
time constraint
duration constraint
time observation

Timers and the U2TP UML 2.0 Profile





timer operations

timer operations

Generics

- MSC-2000
  - parameterizing of messages, instances, data values
- UML 2.0
  - general template mechanism
  - normal value parameters
- Comparison
  - MSC is slightly better

Formal Semantics

- MSC-2000
  - has no Annex B (i.e. formal semantics)
  - but MSC-96 had Annex B (Michel Reniers)
- UML 2.0
  - has no official formal semantics
  - but there are numerous attempts to formalize parts of UML 2.0
    - pUML-group
    - STAIRS
      - formalization of Interactions based on trace semantics and FOKUS-inspired approach
- In practice
  - the tools define the real semantics

Profiling UML

- Profiles in UML is a way to customize UML for a specific purpose
- Official profiles
  - UML Profile for schedulability, performance and time specification
    - still only for UML 1.4
  - UML Profile for Testing
    - U2TP – the first available profile for UML 2.0
    - adds timers and time zones
    - adds data partitioning
    - adds test-specific terms
- Your own profile
  - for some project, or some specific purpose
    - but beware that you add semantics as well as syntax

The Future of MSC/SDL vs UML

- Scenario1: MSC/SDL and UML both prevail
  - they will need different niches, and UML will not let any lucrative niche be left unattended
- Scenario2: UML fails
  - MSC/SDL can thrive in the real-time market
  - something new e.g. from Microsoft outcompetes both
- Scenario3: UML succeeds more
  - MSC/SDL will gradually surrender ground to UML
  - MSC/SDL tool vendors will become UML vendors
  - There may be markets for MSC/SDL profiles of UML
    - what is executable UML?

Which scenario will happen?

- It will not be dependent (purely) on technical reasons