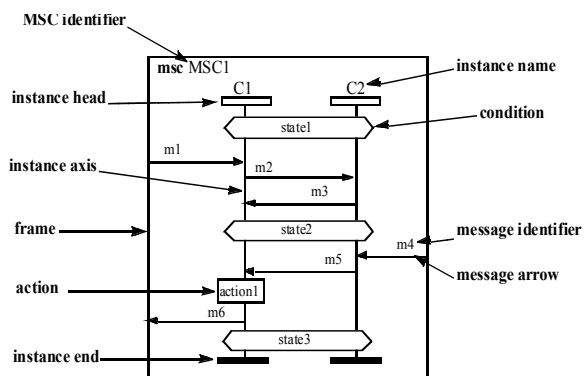


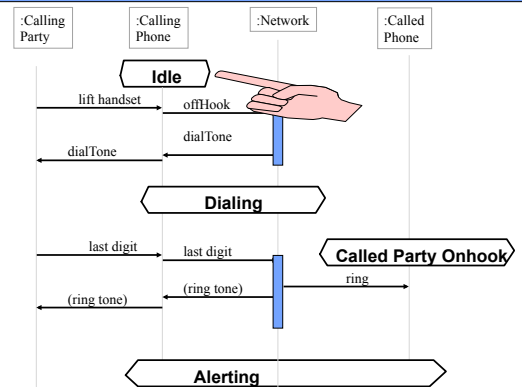
# Interaction Diagrams

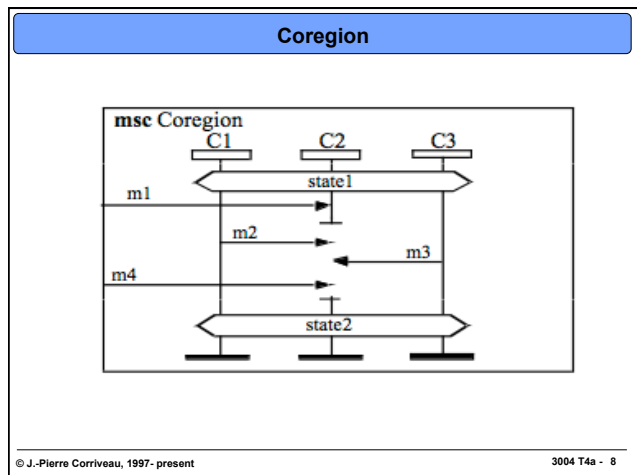
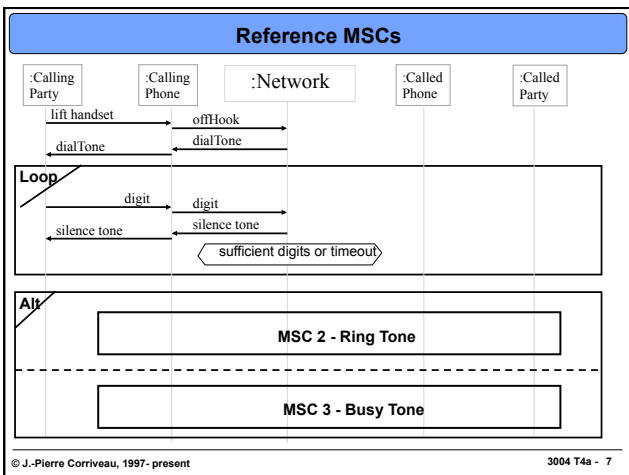
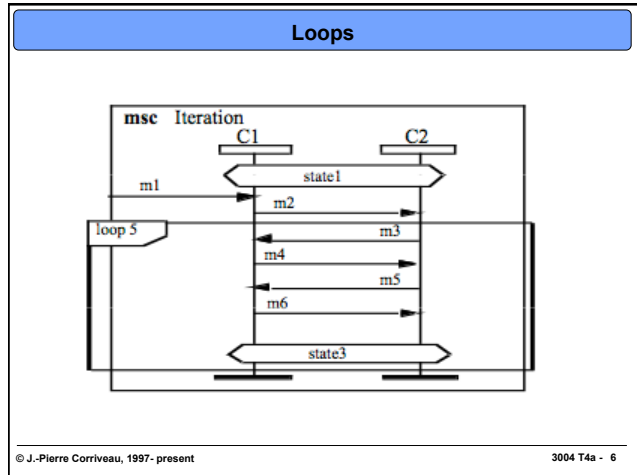
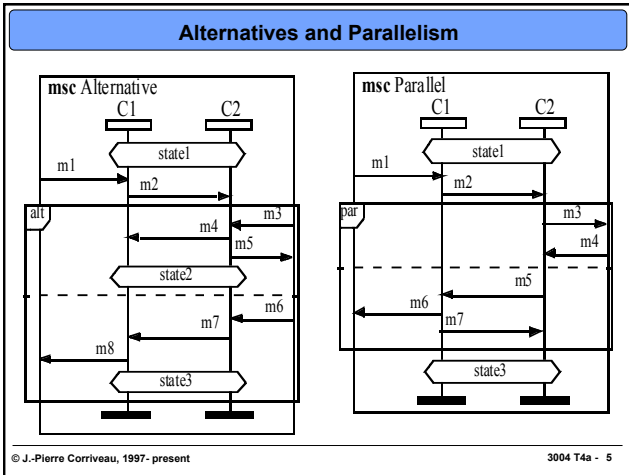
# The Original Notation: Message Sequence Charts (ITU Z.120)

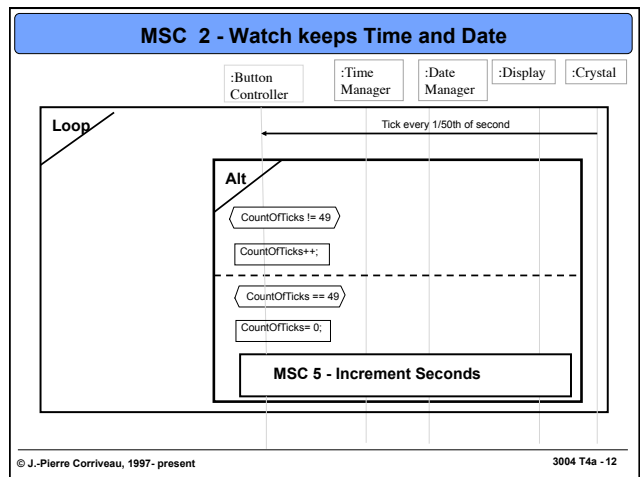
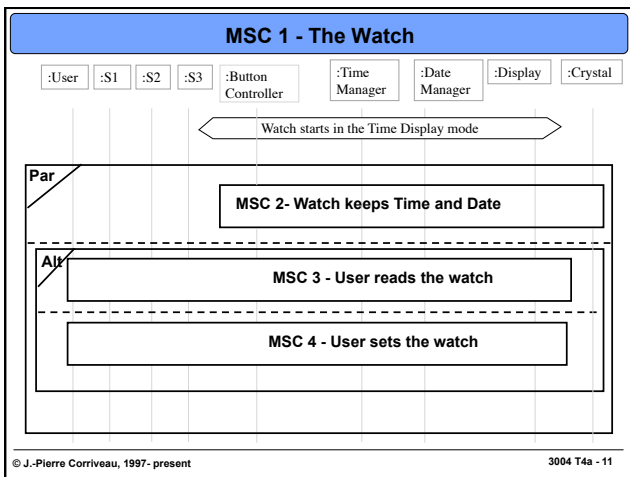
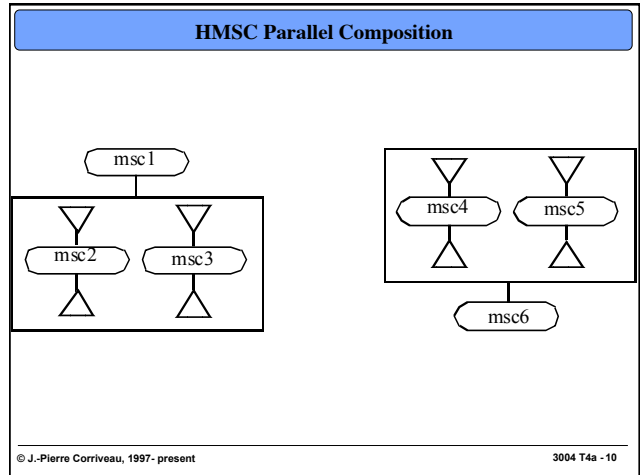
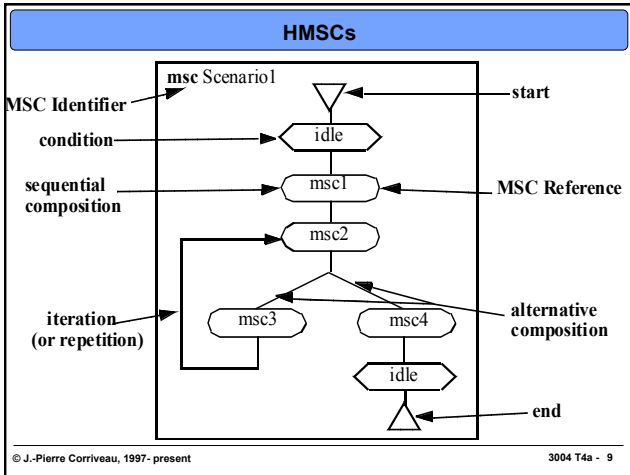
## Basic MSC

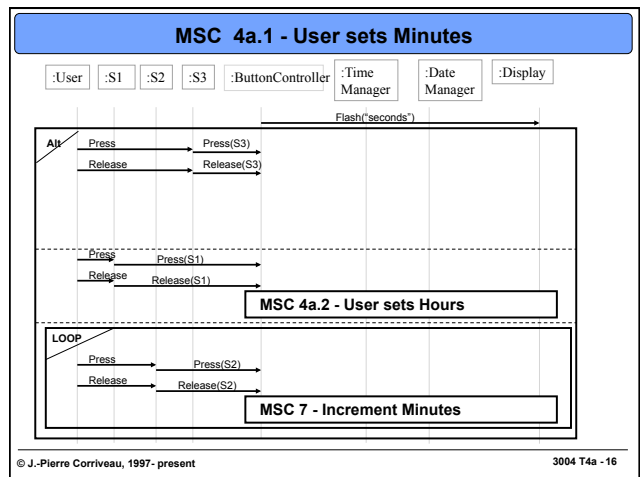
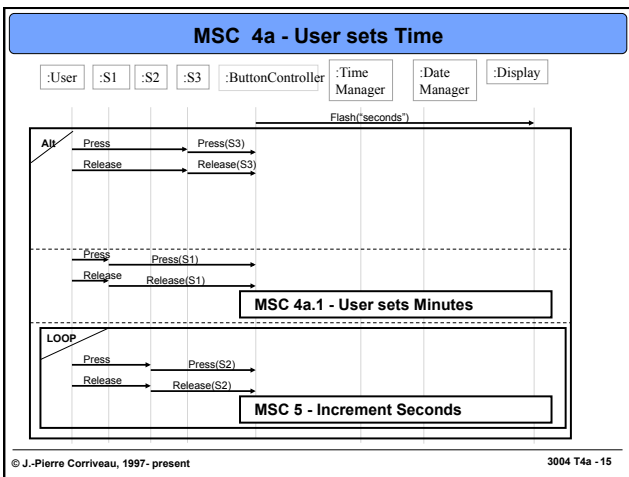
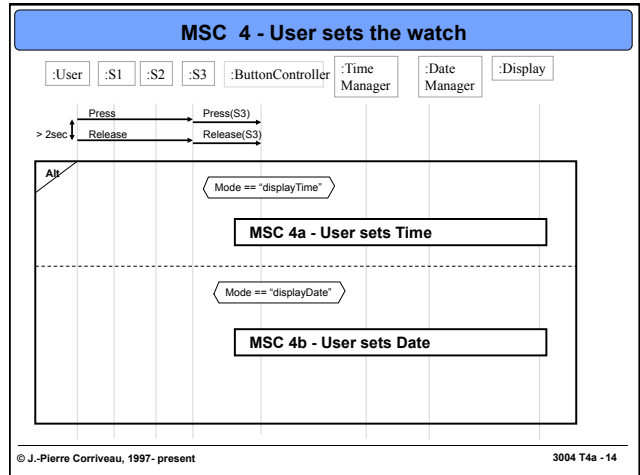
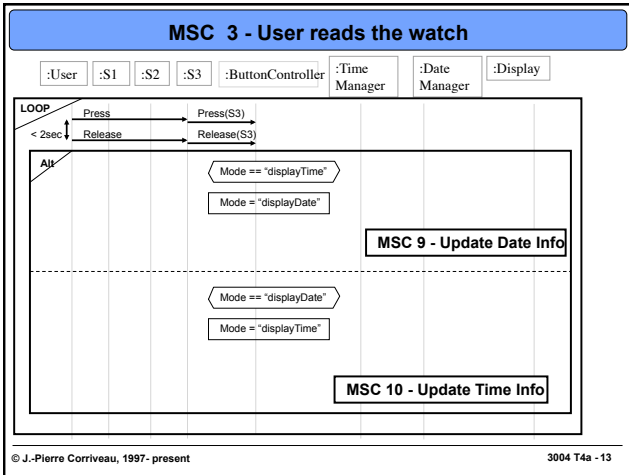


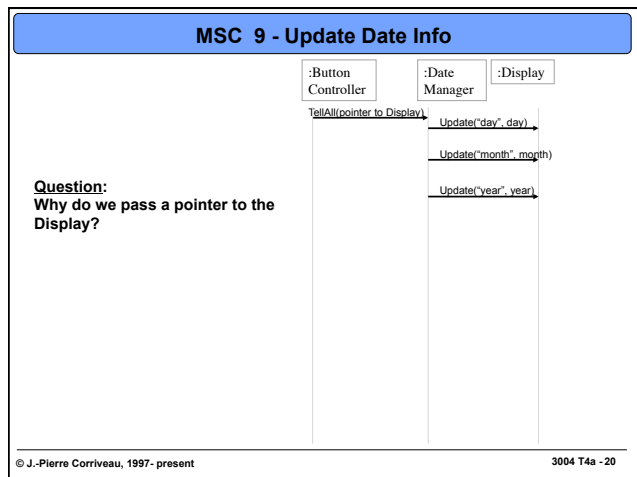
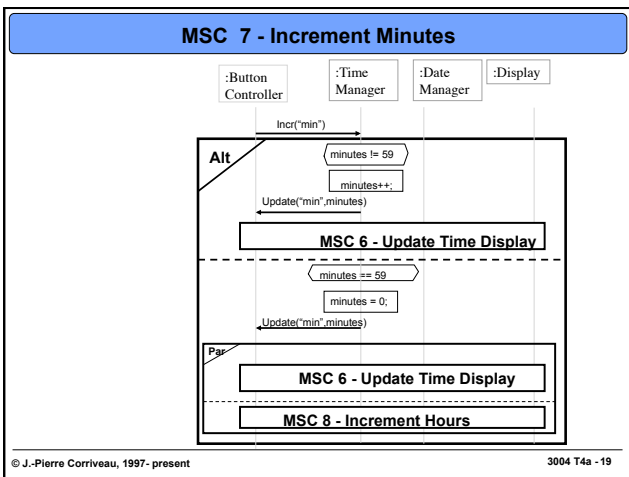
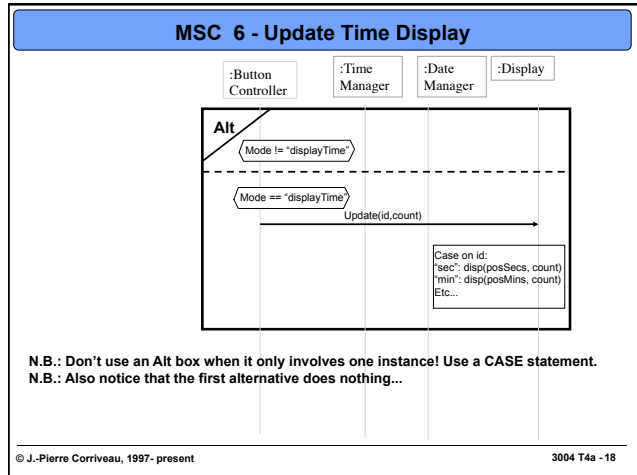
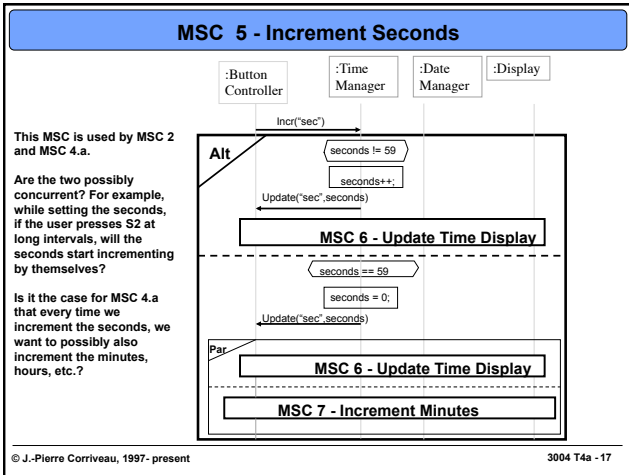
## Constraints (Conditions)

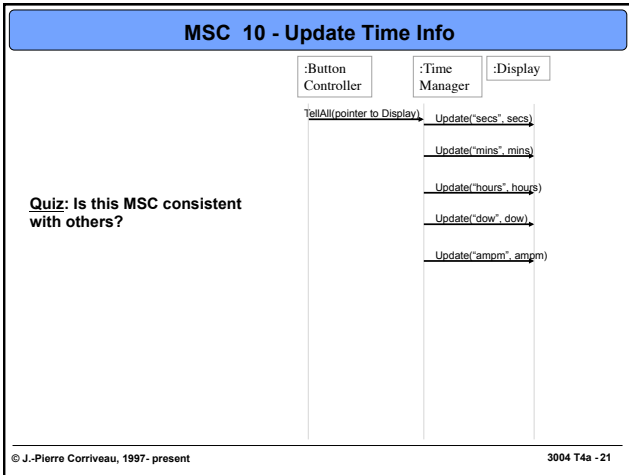












- ### Issues with the MSCs
- **MSC 1:** Have we really addressed the concurrency of the ticks with the rest of the functionality?
  - Is it the case that we want the ButtonController to send the incr("min") message to TimeManager? Why doesn't the latter send it to itself?
  - Are we sure MSC 6 and MSC 7 can be in parallel?
  - **MSC 2** uses MSC 5, which uses MSCs 6 and 7, and so on. Given that there is no user input in the case of the watch keeping the time, this chain makes sense.
    - But MSC 4.a reuses MSC 5, which leads to reusing MSCs 6, 7, etc. Is this what we really want? That is, do we want to increment minutes automatically as we are setting seconds??
    - The alternative consists in NOT incrementing the minutes and cycling back to 0 after reaching 59... So much for reusing MSC 5 and others.
  - **Considering MSC 4.a, if we do set seconds, how do we proceed to setting the minutes?**
    - It seems we are missing a loop around the alternative box.
    - In turn, this suggests cleaning the MSCs to be more generic.
    - The question raised here is how obvious the flow of control between the different MSCs really is. Maybe a high level MSC (i.e., a flowchart) would be more appropriate.
- © J.-Pierre Corriveau, 1997- present 3004 T4a - 22

