Testing Based Modelling

Maarten Damen, Wouter Geurts,
Michel Reniers

Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

Introduction

- Model Based Techniques are
  - valuable in maintenance
  - valuable in proving quality
  - difficult
  - different from current development techniques
  - perceived as additional cost
  - perceived as delaying the project
  - seldom used (certainly regarding legacy)

- => Make models, without difficulty and much cost

Approach

- Luckily there is always a model
  - formal syntax and semantics, but way too detailed.
- IDEA: Match ‘existing model’ with ‘template models’
  - matching criterium: both reproduce visible behaviour
- NEED: Clear visible behaviour
  - use communication logs (for message based system)

Approach (continued)

- ‘template model’ classes
  - periodic behaviour
  - event driven
  - data conversion

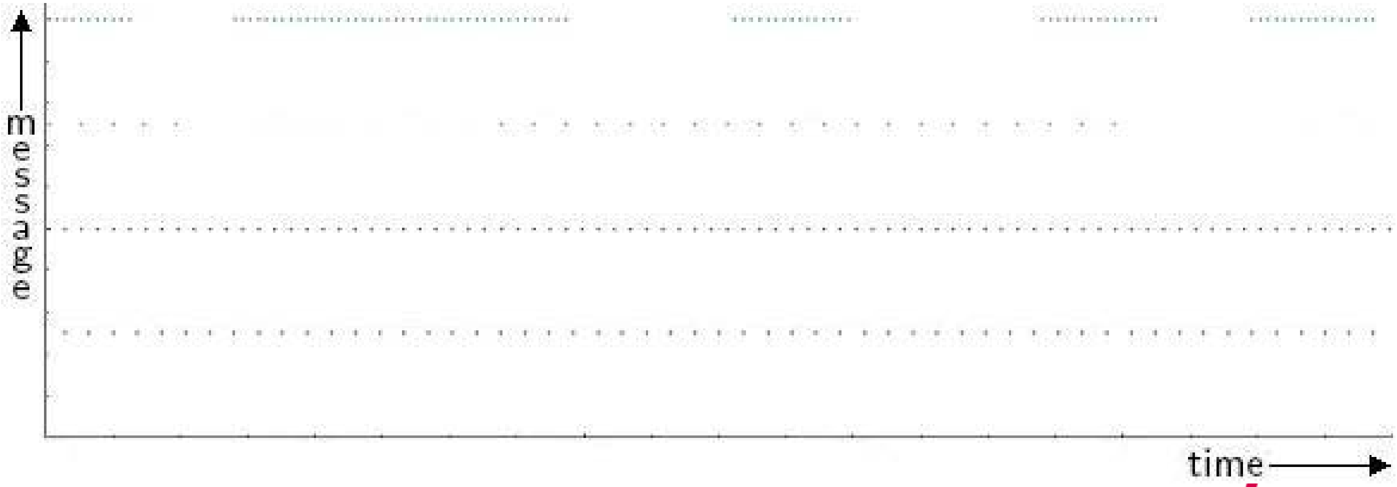
- First try of this approach will follow
  - TU/e, Logica ‘Working Tomorrow’ and ITEA TWINS
  - focus only on the first two bullets

Outline

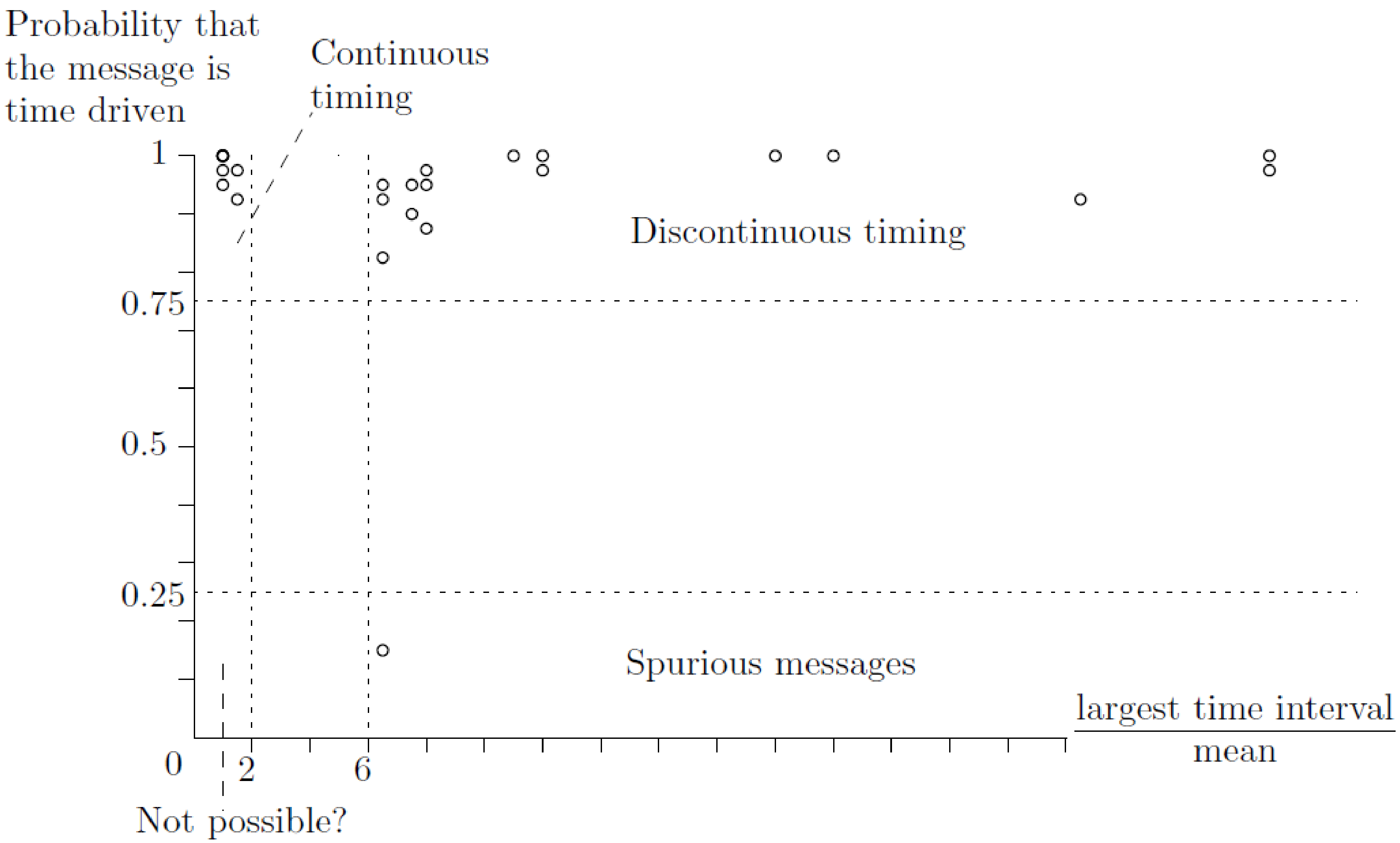
- Introduction
- Approach
- Timing
  - Continuous timing
  - Discontinuous timing
- Spurious behavior
- Conclusions
- Future work
- Questions

Timing

- **Observations:**
  - **Log files are large**
  - **Critical behavior is performed incidentally**
- **Independent of other behavior**
- **Occurrence vs time:**

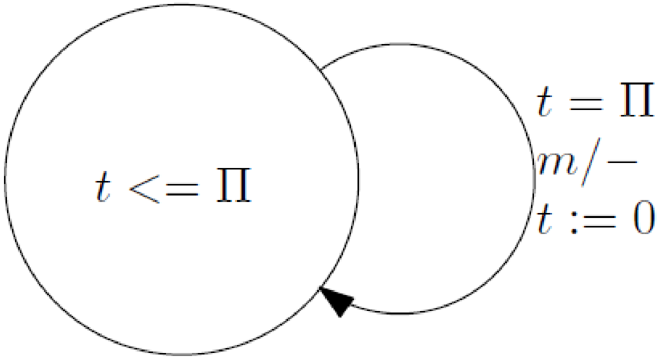


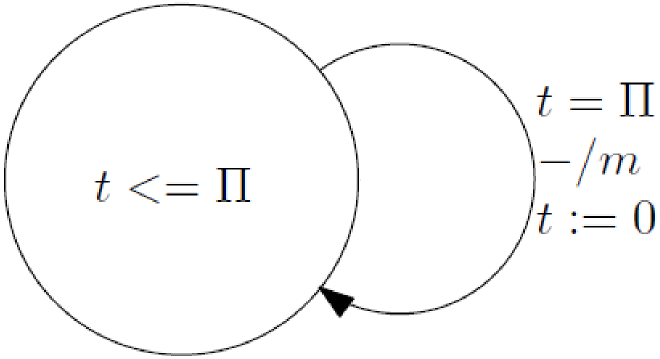
Timing


Probability that
the message is
time driven
Continuous
timing
1
0.75
0.5
0.25
0
2
6
Discontinuous timing
Spurious messages
largest time interval
mean
Not possible?

Continuous timing

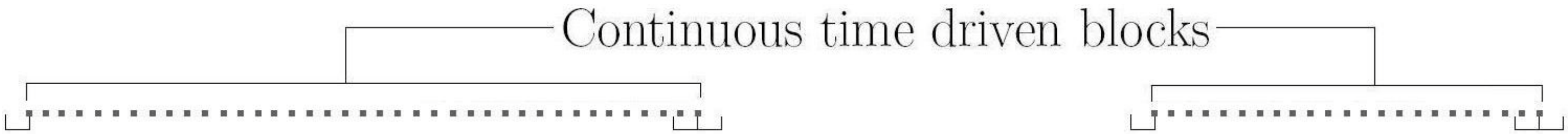
- Time intervals
- All time intervals are of equal size
- Fluctuation
- Missing lines in the log
- Probability
- Model template:





Discontinuous timing

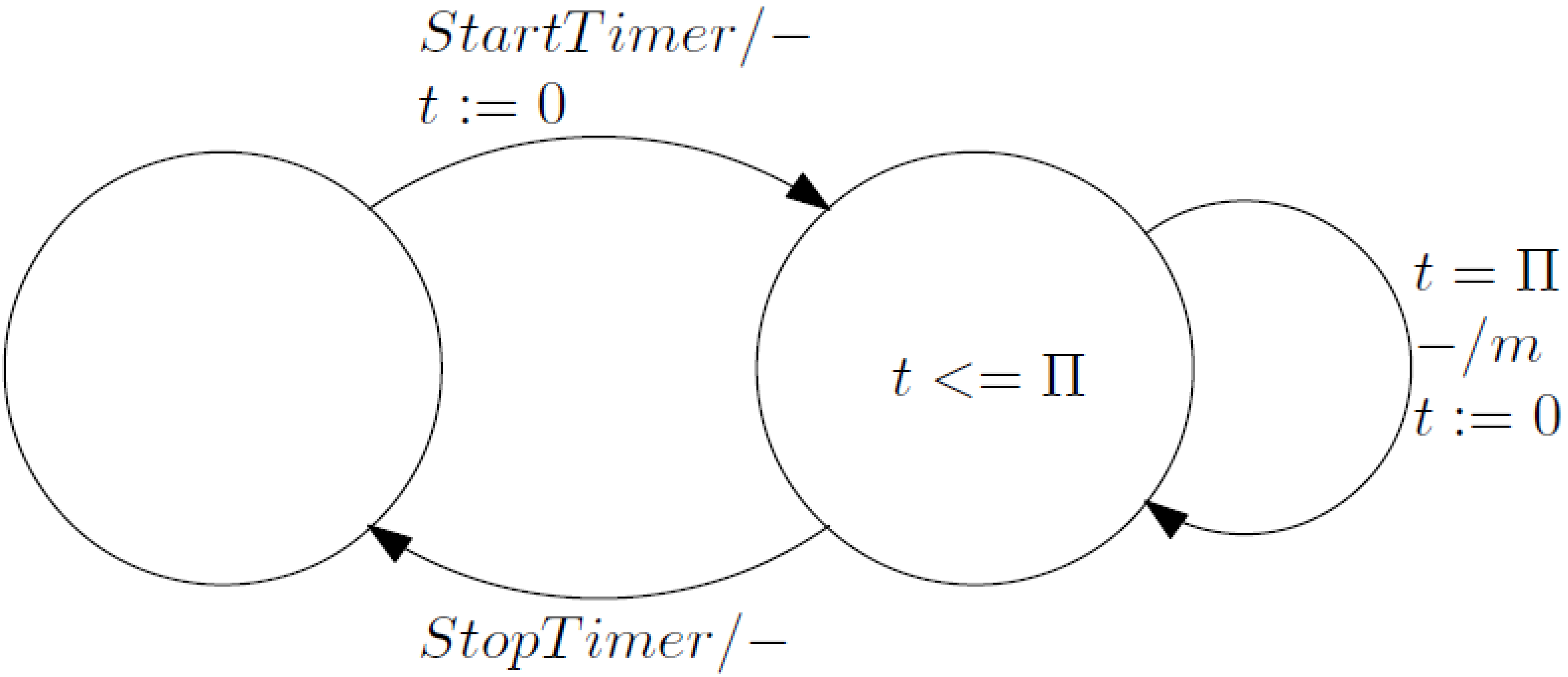
- Some timers can be switched on and off
- Dividing largest time interval by mean value
- Divide and conquer approach



- Determine start/stop messages
- Adding start/stop messages

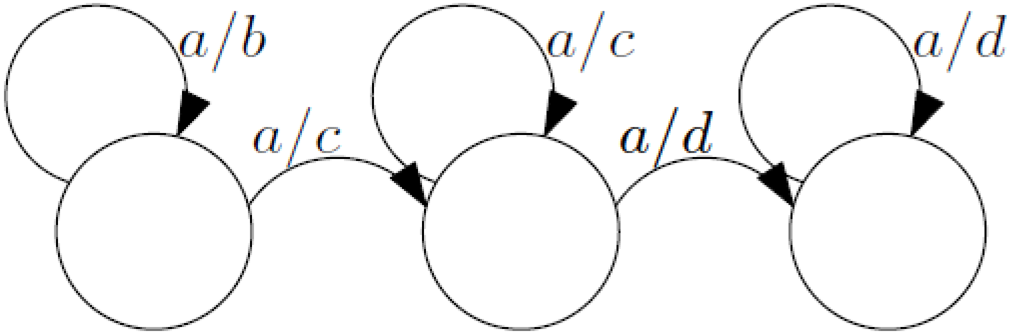
Discontinuous timing

- **Model template:**



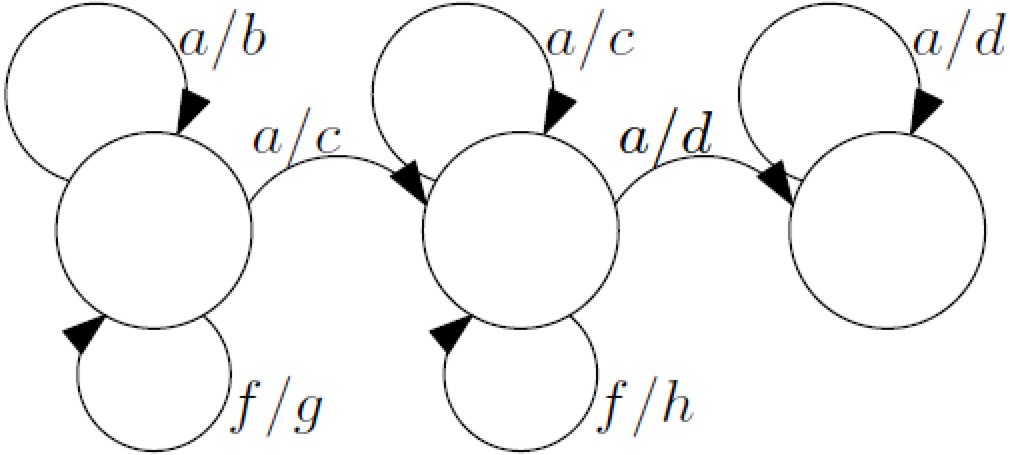
Spurious behavior

- Spurious behavior
- Finding states and transitions
- Different output on same input
- Example trace, a?b!a?c!a?c!a?d!a?d!:



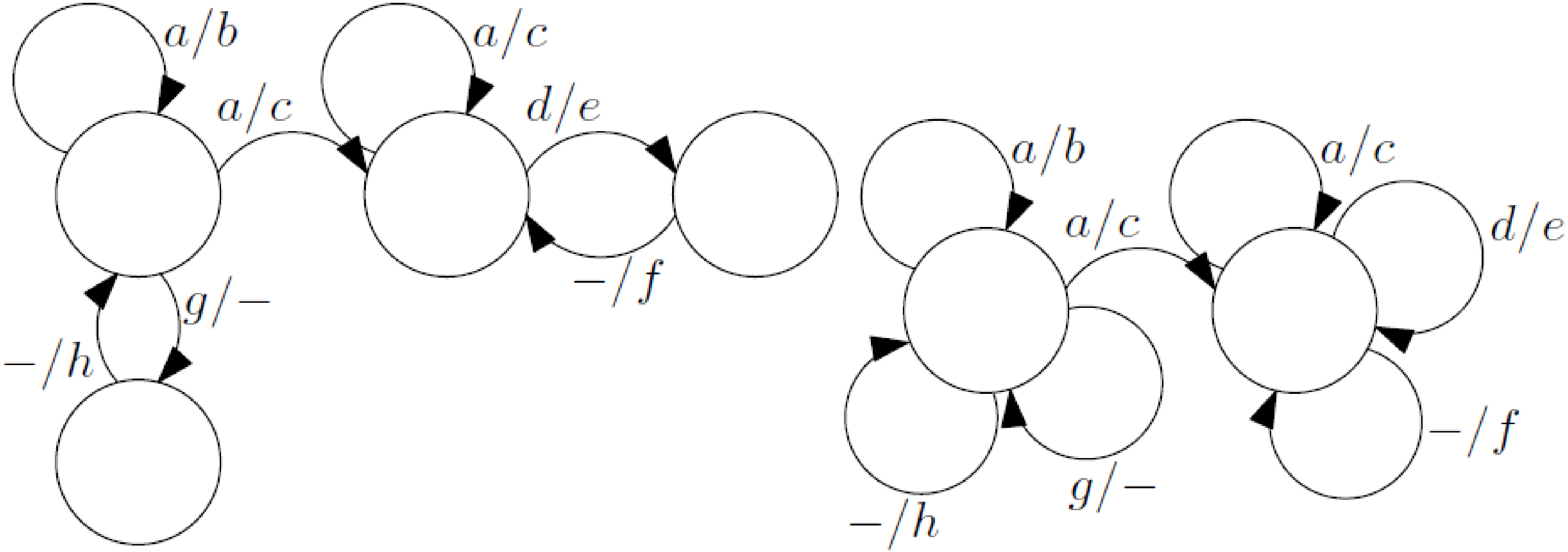
Spurious behavior

- **Process starts over in new state**
- **Example trace, a?b!f?g!a?c!a?c!f?h!a?d!a?d!:**



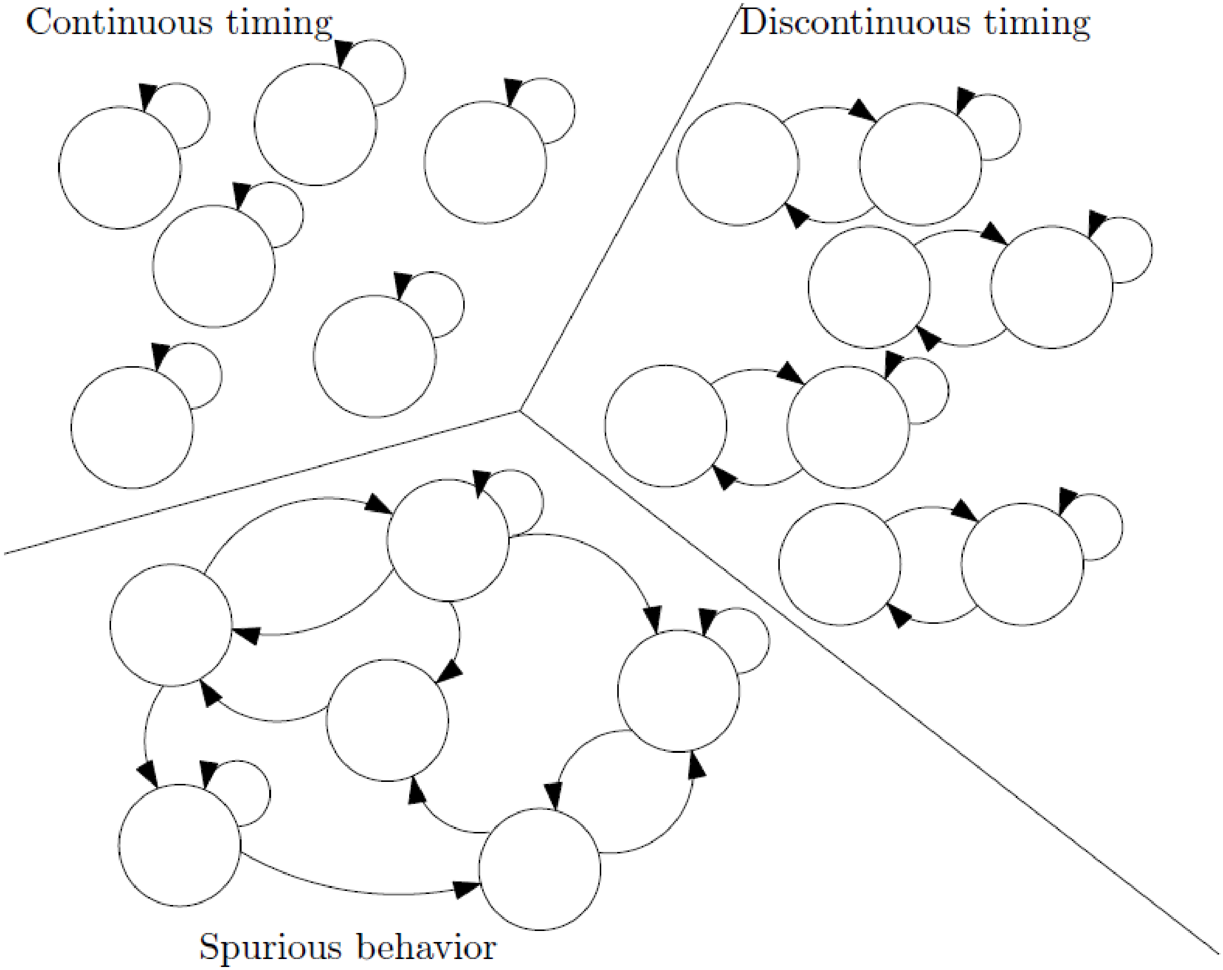
Spurious behavior

- **Handling unrelated outputs**
- **Example trace, a?b!g?<>h!a?c!a?c!d?e!<>f!:**



- **Knowledge of an expert**

Overview


Continuous timing
Discontinuous timing
Spurious behavior

Conclusions

- Finding the original model will be impossible
- Detect timing first
- Start/stop message
- Detecting spurious behavior

Future Work

- Testing on large amounts of data
- Detecting start/stop messages for discontinuous time driven messages
- More timers for the same message
- Changing timer periods
- Determine what messages are reactions to time driven messages
- Continuous time driven blocks
- Detecting a stable state for spurious behavior

Questions