Requirements-based Monitors for Real-Time Systems

Dennis K. Peters
Memorial University of Newfoundland
dpeters@engr.mun.ca
http://www.engr.mun.ca/~dpeters

David L. Parnas
McMaster University

August 22, 2000

Outline

- Motivation
- Four Variable Requirements Model
- Monitors for Real-Time Systems
- Practical Monitors
- Conclusions & Future Work

Motivation

Clear, precise and unambiguous requirements specifications are useful:

- to determine if design is sufficient,
- to check if behaviour is acceptable
  - during system testing (*oracle*),
  - in operation (*supervisor*).

Checking behaviour can be difficult, particularly with real-time requirements.

Monitor: A system that observes and analyses the behaviour of another system (the *target system*).

- What can we conclude from the output of a monitor?
- Are there conditions under which a monitor is infeasible?

The Four Variable Requirements Model

- Distinguish environmental quantities from those that are artifacts of the solution or its description.
- Model quantities as (piecewise-continuous) functions of (continuous) time.
- *Monitored Quantities* — those that should affect the behaviour of the system. (*monitored state function*: $\underline{m}^t \in \mathbf{M}$)
- *Controlled Quantities* — those that the system may be required to change the value of. (*controlled state function*: $\underline{c}^t \in \mathbf{C}$)

System Requirements

For a given $\underline{m}^t$, what are the acceptable values of $\underline{c}^t$?

$\mathbf{REQ} \subseteq \mathbf{M} \times \mathbf{C}$ — the set of acceptable behaviours.

Environmental Constraints

What values of $(\underline{m}^t, \underline{c}^t)$ are possible in the environment?

$\mathbf{NAT} \subseteq \mathbf{M} \times \mathbf{C}$ — the set of possible behaviours.

System Design

What is the behaviour of the input and output devices?

- *Input Quantities* — those program variables available to the software that provide information about $\underline{m}^t$. (Input state function $\underline{i}^t \in \mathbf{I}$.)
- $\mathbf{IN} \subseteq \mathbf{M} \times \mathbf{I}$ — possible values of $\underline{i}^t$ for a given value of $\underline{m}^t$.
- *Output Quantities* — those program variables through which the software can change the value of $\underline{c}^t$. (Output state function $\underline{o}^t \in \mathbf{O}$.)
- $\mathbf{OUT} \subseteq \mathbf{O} \times \mathbf{C}$ — possible values of $\underline{c}^t$ for a given value of $\underline{o}^t$.

Software Requirements

What are the acceptable software behaviours?

$\mathbf{SOFREQ} \subseteq \mathbf{I} \times \mathbf{O}$ —all acceptable software behaviours.

Must ensure that *all* system behaviours are in $\mathbf{REQ}$.

$$\mathbf{SOFREQ} \stackrel{\mathrm{df}}{=}$$

$$\Big\{\left(\underline{i}^t, \underline{o}^t\right) \mid \Big(\forall \underline{m}^t, \underline{c}^t, \left(\mathrm{IN}\left(\underline{m}^t, \underline{i}^t\right) \wedge \mathrm{OUT}\left(\underline{o}^t, \underline{c}^t\right) \wedge \right.$$
$$\left. \mathrm{NAT}\left(\underline{m}^t, \underline{c}^t\right)\right) \Rightarrow \mathrm{REQ}\left(\underline{m}^t, \underline{c}^t\right)\Big)\Big\}$$

Monitors for Real-Time Systems

- Observe the target system behaviour and report if it is consistent, or not, with the requirements.
- Ideal monitor reports value of $\mathrm{REQ}\,(\underline{m}^t, \underline{c}^t)$
- *Safety properties* — if a behaviour violates a property then no extension of it can satisfy it.
  - Monitor can detect violation.
  - Need to re-start after failure.
- *Liveness properties* — any finite duration behaviour can be extended such that it satisfies the property.
  - Can't be detected by monitor.
  - Usually not strong enough for real requirements.

Monitor Configuration

Monitors are systems too!

- *Monitor software inputs* — program variables available to the monitor software that relate to $(\underline{m}^t, \underline{c}^t)$. (Monitor input state function $\underline{s}^t \in \mathbf{S}$.)
- $\mathbf{IN_{mon}} \subseteq (\mathbf{M} \times \mathbf{C}) \times \mathbf{S}$ — possible values of $\underline{s}^t$ for given $(\underline{m}^t, \underline{c}^t)$.
- *Software monitor* — directly observes target system input and output quantities $(\underline{i}^t, \underline{o}^t)$.

$$\mathbf{IN_{mon}} \quad = \quad \left\{\left(\left(\underline{m}^t, \underline{c}^t\right), \left(\underline{i}^t, \underline{o}^t\right)\right) \middle| \mathrm{IN}\left(\underline{m}^t, \underline{i}^t\right) \quad \wedge \quad \mathrm{OUT}\left(\underline{o}^t, \underline{c}^t\right)\right\}$$

- *System monitor* — observes $(\underline{m}^t, \underline{c}^t)$ using its own input devices.

Monitor input devices usually aren't perfect — $\mathbf{IN_{mon}}$ is non-functional.

- *pessimistic monitor* — Requires that *all* behaviours that could have resulted in a given $\underline{s}^t$ are in $\mathbf{REQ}$.

$$\mathbf{MON_{pe}} \stackrel{\mathrm{df}}{=}$$

$$\Big\{\underline{s}^t \in range(\mathbf{IN_{mon}}) \Big| \Big(\forall \left(\underline{m}^t, \underline{c}^t\right) \in \mathbf{M} \times \mathbf{C},$$

$$\left(\mathrm{IN_{mon}}\left(\left(\underline{m}^t, \underline{c}^t\right), \underline{s}^t\right) \wedge \mathrm{NAT}\left(\underline{m}^t, \underline{c}^t\right)\right)$$

$$\Rightarrow \mathrm{REQ}\left(\underline{m}^t, \underline{c}^t\right)\Big)\Big\}$$

- For software monitor $\mathbf{MON_{pe}} = \mathbf{SOFREQ}$.
- If $\mathbf{MON_{pe}} = \emptyset$ then monitor is *infeasible*.

- *optimistic monitor* — *some* behaviour that could have resulted in a given $\underline{s}^t$ is in $\mathbf{REQ}$.

$$\mathbf{MON_{op}} \stackrel{\mathrm{df}}{=}$$
$$\left\{\underline{s}^t \in range(\mathbf{IN_{mon}}) \middle| \left(\exists \left(\underline{m}^t, \underline{c}^t\right) \in \mathbf{M} \times \mathbf{C}, \right.\right.$$
$$\mathrm{IN_{mon}}\left(\left(\underline{m}^t, \underline{c}^t\right), \underline{s}^t\right) \wedge \mathrm{NAT}\left(\underline{m}^t, \underline{c}^t\right)$$
$$\left.\left. \wedge\, \mathrm{REQ}\left(\underline{m}^t, \underline{c}^t\right)\right)\right\}$$

- Difference between $\mathbf{MON_{pe}}$ and $\mathbf{MON_{op}}$ is indicative of appropriateness of monitor input devices.
- If $\mathbf{IN_{mon}}$ and ${\mathbf{IN_{mon}}}^{-1}$ are both functions then $\mathbf{MON_{pe}} = \mathbf{MON_{op}}$ (Ideal monitor).

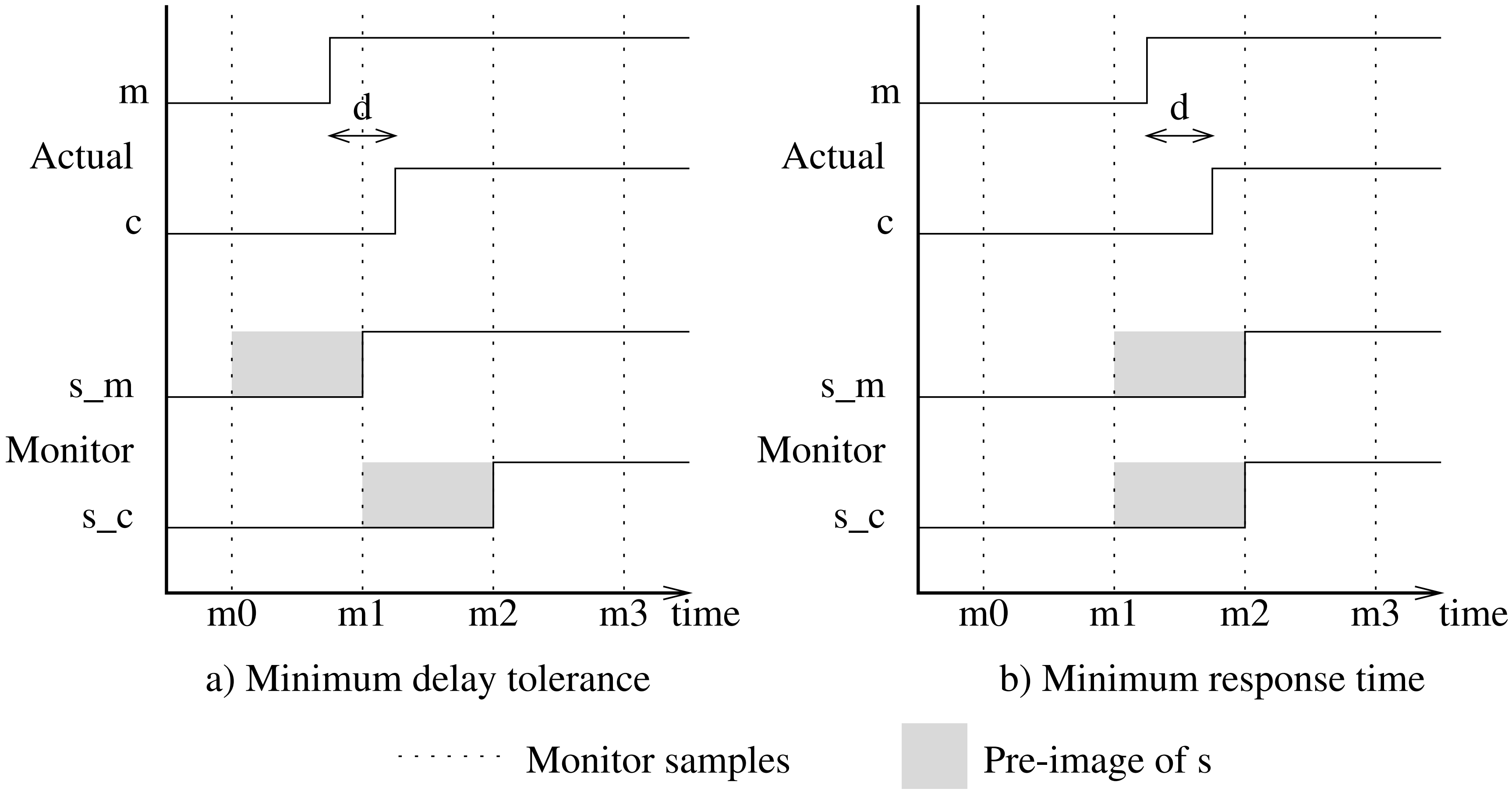
Practical Monitors
Accuracy

- *False negatives*, $\mathbf{FN}$ — acceptable behaviours reported as unacceptable.
  $\mathbf{FN} \stackrel{\mathrm{df}}{=} \mathbf{REQ} \cap \mathbf{NEG}$, where
  $$\begin{aligned} \mathbf{NEG} &= \text{behaviours reported as unacceptable (pessimistic)} \\ &= \text{ image of } \overline{\mathbf{MON_{pe}}} \text{ under } \mathbf{IN_{mon}}^{-1} \end{aligned}$$

- *False positives*, $\mathbf{FP}$ — unacceptable behaviours reported as acceptable.
  $\mathbf{FP} \stackrel{\mathrm{df}}{=} \overline{\mathbf{REQ}} \cap \mathbf{POS}$, where
  $$\begin{aligned} \mathbf{POS} &= \text{behaviours reported as acceptable (optimistic)} \\ &= \text{image of} \mathbf{MON_{op}} \text{under} \mathbf{IN_{mon}}^{-1} \end{aligned}$$

- Both $\mathbf{FN}$ and $\mathbf{FP}$ are determined by $\mathbf{IN_{mon}}$ and $\mathbf{REQ}$.

Discrete Time



Condition 1. $\delta_{mon} < \frac{1}{2}min(\mathbf{Delay})$, *where* $\mathbf{Delay}$ *is the set of maximum delay tolerances for the dependent quantities.*

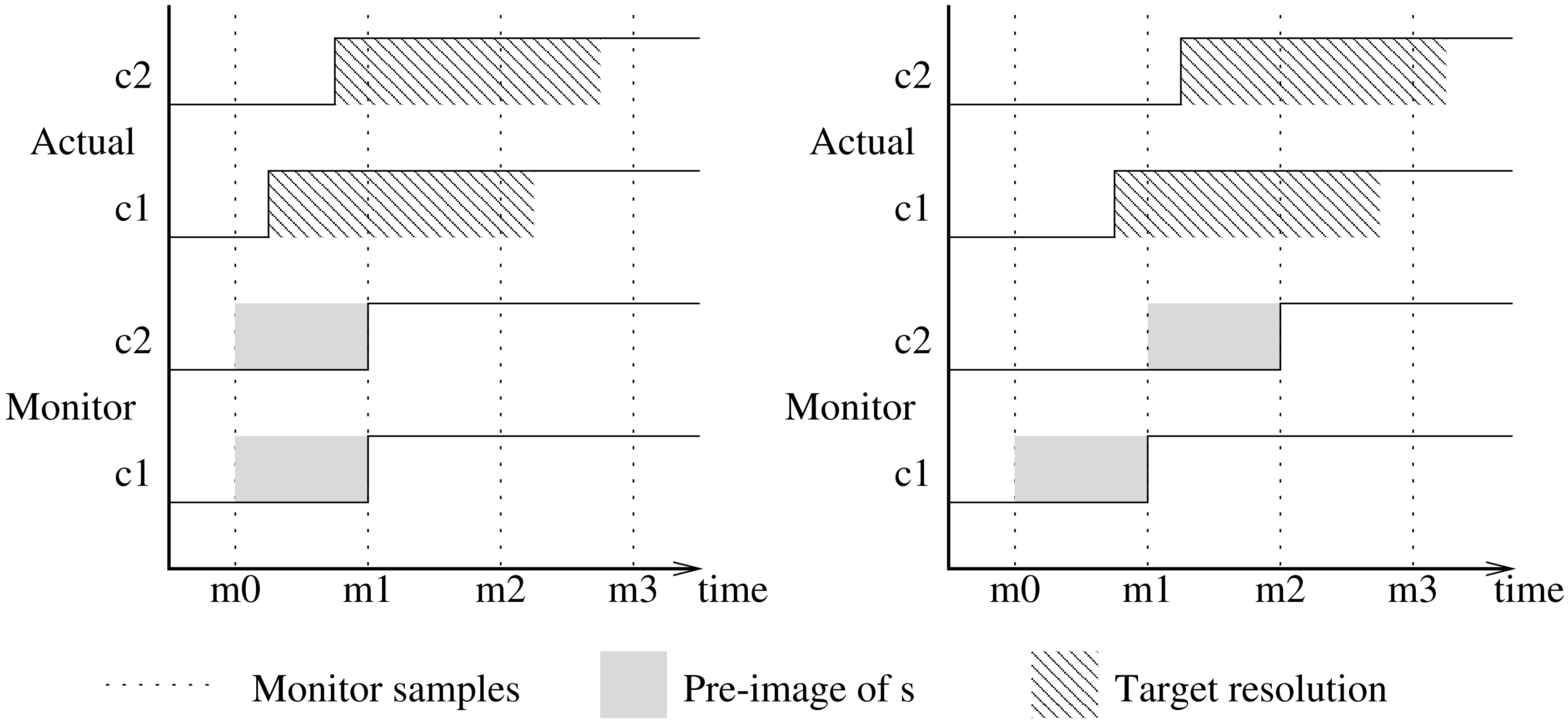
Condition 2. $\delta_{mon} < min($*response time*$)$

Condition 3. *monitor measurement error* $<$ *tolerance*

- Conditions 1 and 3 are necessary for monitor feasibility.
- Condition 2 prevents false negatives from too quick system response.

Non-determinism

- To permit implementation using discrete time systems, $\mathbf{REQ}$ must allow events within the *time resolution* to be considered either simultaneous or non-simultaneous.
- Monitor software must allow non-determinism.
- Software monitor 'sees' the same events, so it can assume determinism.


c2
Actual
c1
c2
Monitor
c1
m0
m1
m2
m3
time
c2
Actual
c1
c2
Monitor
c1
m0
m1
m2
m3
time
Monitor samples
Pre-image of s
Target resolution

Conclusions

- Monitors can be used to check safety properties, but not liveness.
- Usefulness and feasibility of monitors depends on monitor input devices and $\mathbf{REQ}$.
- Monitors can be used as oracles or for supervision.

Future Work

- More target systems.
- How to specify input and output devices (formally and readably)?
- Sufficient conditions for feasibility (or stronger)?