A new Programming Model for Dependable Adaptive Real-Time Applications

Presented by
António Casimiro

48th Meeting of IFIP Working Group 10.4
Hakone, Japan, July 1-5, 2005

Context

- Work developed in CORTEX, in which the concept of sentient objects was introduced
  - Autonomous entities with sentience (e.g. robots)
  - Geographical dispersion
  - Real-time & safety requirements
  - Availability
- Several issues addressed in CORTEX
  - Programming model for sentient applications
  - Interaction model
  - WAN-of-CANs architecture (systems-of-systems)

Dealing with uncertainty

- We defined a generic approach to reconcile uncertainty with the need for predictability
- This could be (and was) applied in CORTEX, for sentient applications
- Make the application behave [safely, timely, securely, etc] in the measure of what can be expected from the environment
- Provide guarantees in the way that is done

⇒ Dependable adaptation

Back to the roots

- Initial idea proposed in 1999
  - Formal definition of the relevant properties:
    - No-contamination
    - Coverage stability
  - Definition of approaches for dependable application programming:
    - Fail-safe approach (fail-safe applications)
    - Reconfiguration & adaptation (time-elastic, t-safe apps)
    - Replication

Meanwhile…

During the course of CORTEX

Programming principle

- General and systematic approach:
  - QoS coverage service
    - The user simply provides the needed coverage
    - The service indicates the bound that must be used
    - For applications with time-safety and time-elasticity
  - Timing failure detection service
    - The user provides a bound for some action
    - The service will execute an handler upon failure detection

Making it dependable

- To adapt the QoS it is necessary to:
  - monitor the actual QoS being provided
  - decide if adaptation is necessary

- To dependably adapt the QoS we must:
  - observe the environment in a dependable way
  - apply a rigorous strategy to decide when and how to adapt

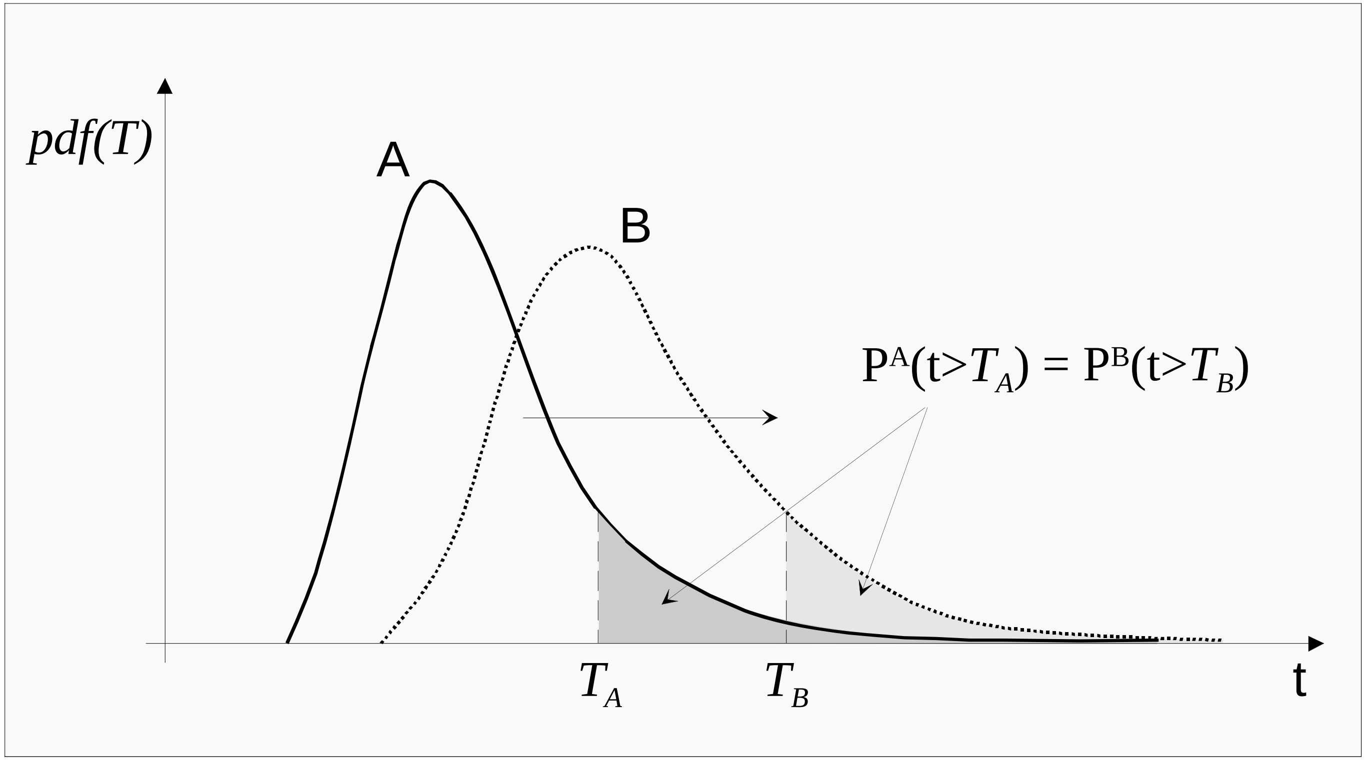
Dependable adaptation

- First, it is necessary to trust the service that provides the measurements (durations)

  - in the value domain (correct measurements)…
  - …and in the time domain (timely measurements)

Dependable adaptation

- Then, decide when and how to adapt

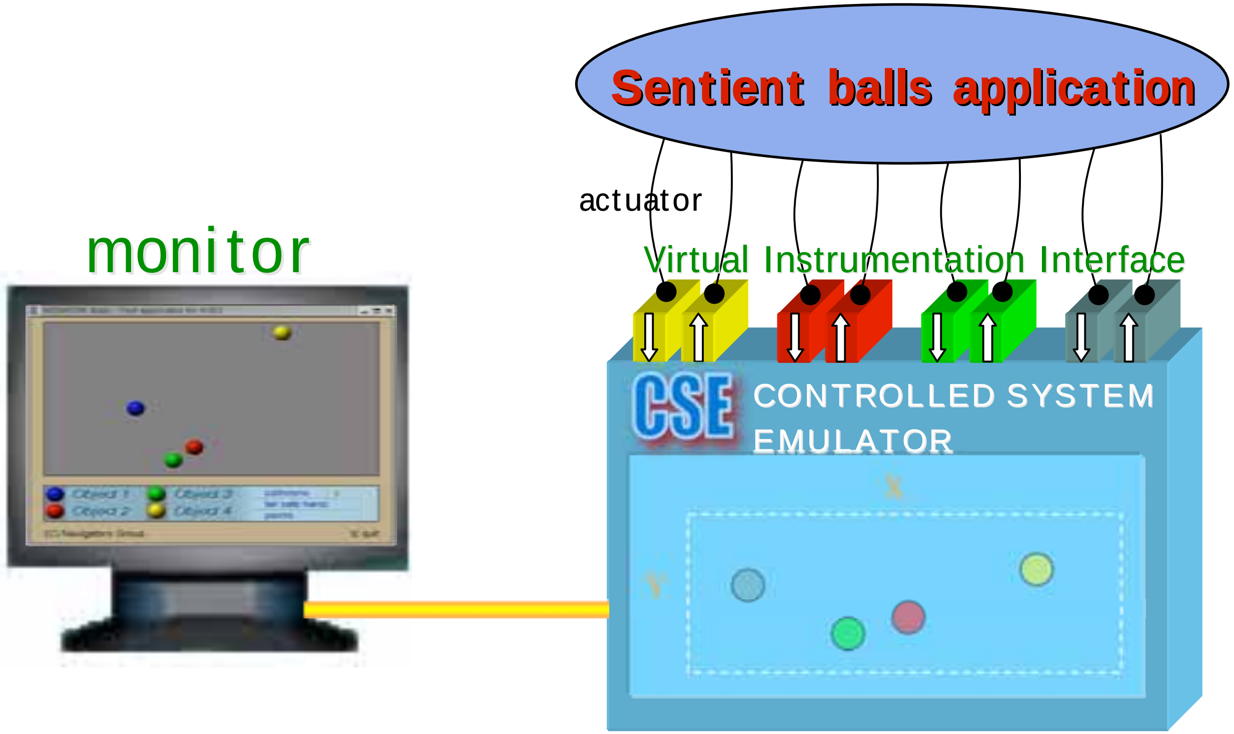


Finally…

We applied the programming model

Sentient balls application

- Physical environment is emulated



Emulator

- Emulated environment: four entities shaped as colored balls move in a space with a certain speed and direction
- A Virtual Instrumentation Interface allows to:
  - acquire ball positions, directions and speeds;
  - change ball movement (speed and direction)
- The sentient application (ball controllers) uses the TCB for the underlying services:
  - QoS Adaptation
  - Timing Failure Detection

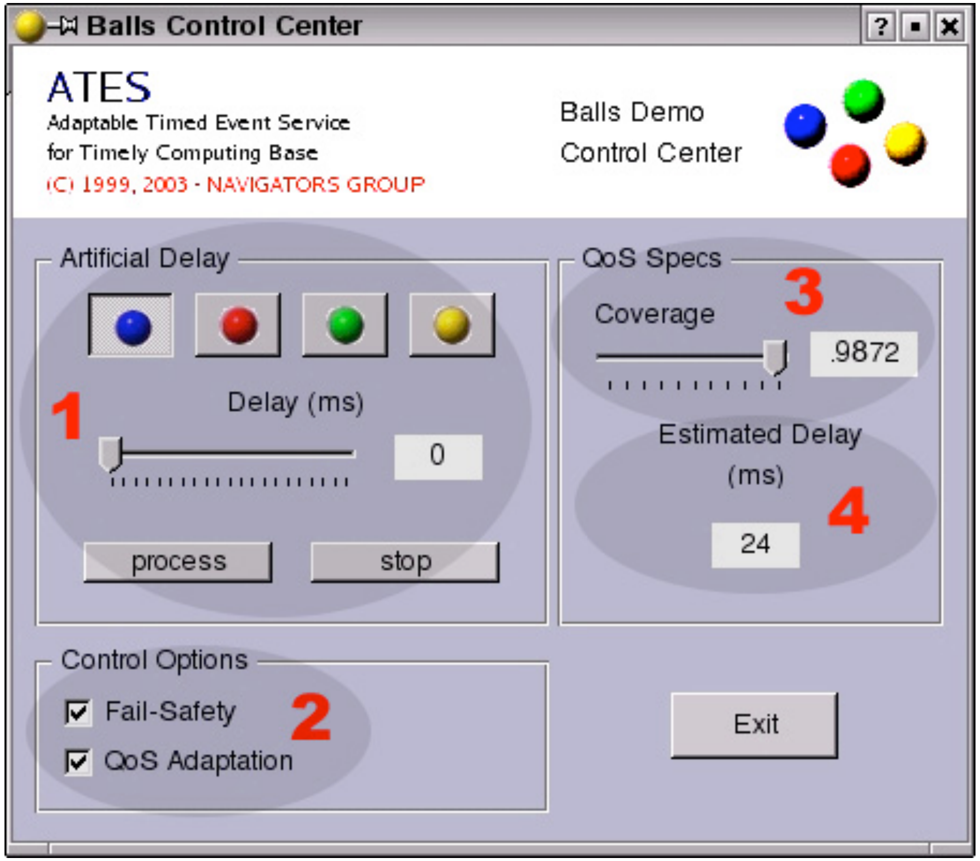
Fail-Safety Demo

- When Fail-Safety is ON:
  - Delivery delay of events is controlled using the TCB distributed TFD
  - Timing failure detected ➔ stop balls in timely way
- When Fail-Safety is OFF:
  - Timing failures can cause balls to crash!

QoS-Adaptation Demo

- When QoS-Adaptation is ON:
  - The service indicates the estimated delay that corresponds to requested coverage value
  - This value is used to determine and set ball speed that preserves safety
  - Coverage stability is achieved
- When QoS-Adaptation is OFF:
  - No speed adaptation takes place
  - Assumed delay keeps constant, possibly leading to coverage degradation due to timing failures

A small taste of it…



Where is the paper?

- MAIN FEATURE of May 2005 issue of IEEE Distributed Systems On-Line Journal:
  - http://dsonline.computer.org
  - http://dsonline.computer.org/portal/site/dsonline/menuitem.9ed3d9924aeb0dcd82ccc6716bbe36ec/index.jsp?&pName=dso_level1&path=dsonline/0505&file=o5001.xml&xsl=article.xsl&
- *A New Programming Model for Dependable Adaptive Real-Time Applications*
  Pedro Martins, Paulo Sousa, António Casimiro, Paulo Veríssimo
  IEEE Distributed Systems Online, vol. 6, no. 5, 2005.

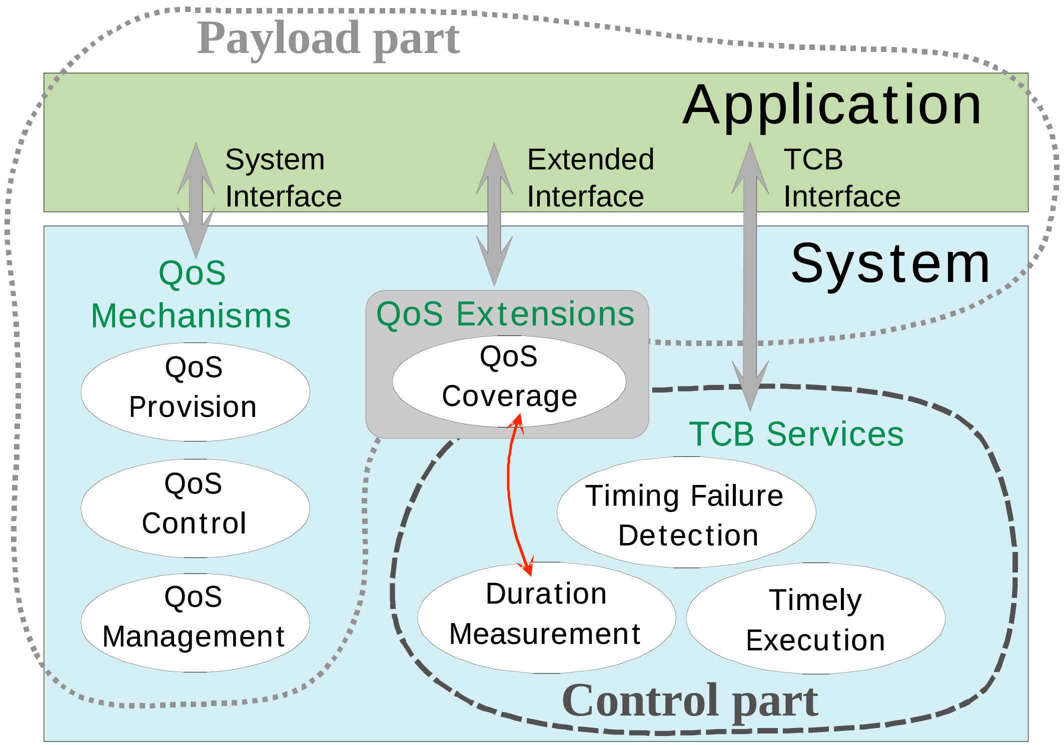
you may also get there from our web site, www.navigators.di.fc.ul.pt under "Recent Documents".

…a small movie

Extra slides

QoS coverage service

In a system with a TCB



Implementation

- We use a known result from prob. theory:

$$P(D > t) \leq \frac{V(D)}{V(D) + (t - E(D))^2}, \text{ for all } t > E(D)$$

  - which allows the calculation of an upper bound for the probability of a time bound t being violated

- Given the coverage $C_{\min}$, t is obtained with:

$$t = \frac{2E(D) + \sqrt{4E(D)^2 - 4(E(D)^2 + V(D) - \frac{V(D)}{1 - C_{\min}})}}{2}$$

Implementation issues

- Estimation of Expected value and Variance
  - E(D) and V(D) correspond to the average and variance of a set of values obtained during an interval of mission
  - The size of the set depends on the application

- Contributing factors for accuracy loss:
  - Error associated to the measured durations
  - Error introduced by the estimation (finite number of samples)
  - Error that results from using an upper bound for the probability

- Results can be improved by reducing errors:
  - Measure durations with smaller errors
  - Get rid of pessimistic assumptions (e.g. no recognition abilities)