Real-Time Computing Technology in the 21st Century

IEEE Distinguished Visitor Lecture

Professor Aloysius K. Mok
Department of Computer Sciences
The University of Texas at Austin

What Is a Real-Time System?

The Space Shuttle as a Real-Time System

- The Space Shuttle cannot go into space without continuous real-time computer control
  - Vehicle is controlled by a group of four computers, with a fifth backup that is programmed by an independent team of programmers
  - There are more computers in the mission control center on the ground that are responsible for monitoring the Space Shuttle

You Don't Need to Go to Space to Find a Real-Time System

Timeliness requirements are found in many systems:

- Cell phones
- Electric power grids
- Medical devices
- Video games
- Stock trading systems

Real-time computing technology
is a critical cross-cutting technology

A Misconception: Real-Time = Fast?

- A real-time system must provide response in a ***fast enough*** manner, e.g., within a specified hard/soft deadline
- How fast is fast enough?
  - Relatively easy to answer for traditional control systems
  - Difficult to answer for the sophisticated systems of the future
- ***The computer system managing the Space Shuttle Challenger was not fast enough to detect anomalous dynamic pressure on the structure of the vehicle before disaster struck***

How Do We Ensure Timely Response?

Need technologies for:

- Specification of timeliness requirements
- Validation and verification to ensure timing correctness
- Resource allocation and scheduling to meet timeliness requirements

Precise Specification Can be Tricky

"**Reset and resynchronize the timer periodically once an hour**"

Interpretations:

1. Resynchronize exactly on the hour, at 8, 9, 10, …
2. Resynchronize once between 8 and 9, between 9 and 10, between 10 and 11, …
3. Every resynchronization must be within an hour of the last one.

Which interpretation would you choose, and does it matter?

Imprecise Specification Can be Deadly

Quoted from the LA Times:

> " Two separate human errors caused a breakdown in radio communications around 4:30 pm on 14 Sep 2004 that lasted for three hours and brought Southern California's major airports to a near-stop. There were at least five instances in which planes came too close during the first 15 minutes of the communications breakdown.
>
> The FAA's radio system in Palmdale shut itself down because a technician failed to reset an internal clock **-- a routine maintenance procedure required every 30 days** by the FAA. "
>
> " When the system was upgraded about a year ago, the original [Voice Switching Communications System] computers were replaced by Dell computers using Microsoft software. Baggett said the Microsoft software contained an internal clock designed to shut the system down after 49.7 days to prevent it from becoming overloaded with data."

Quoted from an air traffic controller in comp.risks:

> " Basically, the system needs to be reset about once a month- or more specifically, once every 30 days or so. I heard a rumor that part of the problem in LA was that they'd done the reset at the beginning of August, but had put it off for September... and were planning to do it at the end of the month."

For Precision, Let Us Try Logic

Suppose **p** denotes the proposition:

"The airplane is on the ground"

Is **p** true or false?

The answer of course, depends on the context (system state). The airplane may be on the ground at 7:00 a.m. but may have taken off at 7:30 a.m.

For computer system execution, the universe of discourse is not an unchanging system state but an evolvement (trajectory) of states.

Logics and Formal Methods

- First-order logic is "static" in that a proposition is either true or false in the universe of discourse. To reason about a dynamic universe, mathematical devices have been invented to allow us to reason about sequences of system states.
- Specialized logics have been invented for precise specification of timing properties of systems, e.g., CTL (Computational Tree Logic), LTL (Linear Temporal Logic). Temporal logics are mathematical devices for reasoning about sequences.

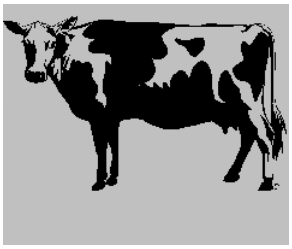
Logics and Formal Methods

- Tremendous advance has been achieved in the last 20 years in automating the verification of system properties by various mathematical methods; the most notable method is *model checking* which is now routinely used in hardware design by companies such as Intel. Progress is being made to apply these techniques to verify real-time, safety-critical application software. We are not there yet.
- There remains, however, the unavoidable problem of translating vague user expectations into the precise language of logic

Garbage in, garbage out!

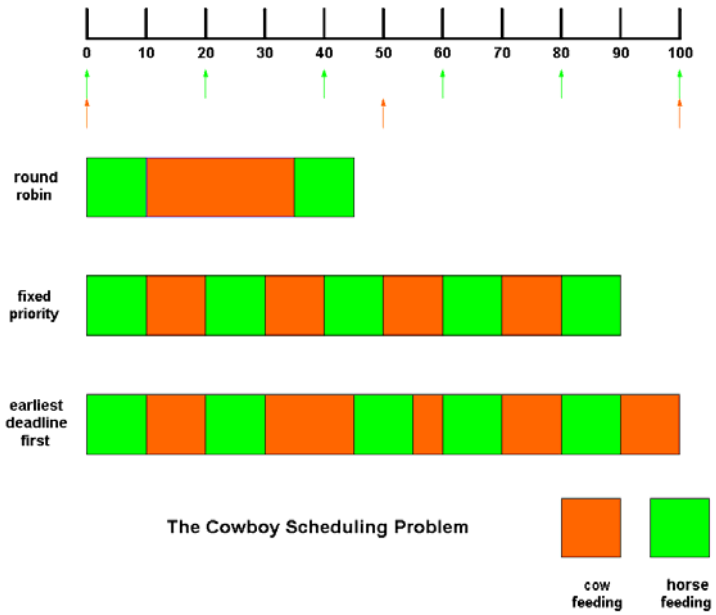
Real-Time Scheduling

Feed horse periodically every 20 minutes for 10 minutes each period



Feed cow periodically every 50 minutes for 25 minutes each period

The Cowboy Scheduling Problem



Real-Time Scheduling Policies

1. Round-robin scheduler
2. Fixed-priority scheduler
3. Earliest deadline scheduler

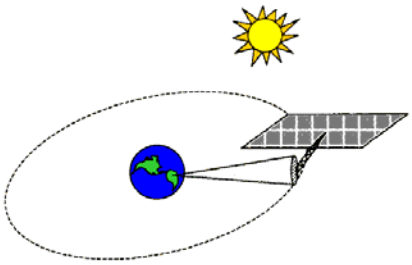
- Which scheduler is guaranteed to work?
- Given a scheduler, how do we know that it will always work?
- Is there a "best" scheduler?

Seminal work in Liu & Layland's 1973 JACM paper

Many Technical Challenges for the 21st Century Real-Time System Designer

- Real-time systems in the 21st century will be
  - Bigger
  - More complex
  - More safety-critical
  - More Internet-dependent

Bigger!

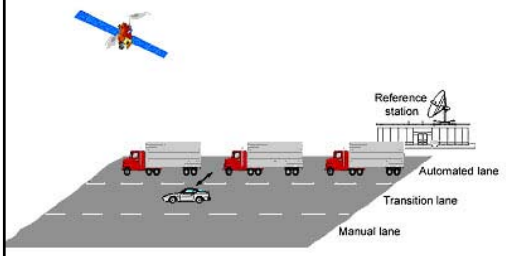


Micro satellites

- Tens of thousands of micro (a few pounds) satellites
- Can form configurations to simulate large lenses to see far into space
- Can provide versatile allocation of communication bandwidth

Interesting real-time coordination problems whose solution must be able to scale easily

More Complex!



Automated Highway

- Sensors embedded in and alongside highway
- Every car interfaces with highway sensor system and takes command from system
- Must work in all weather

Highly heterogeneous and open system environment with distributed, dynamic real-time requirements

More Safety Critical!

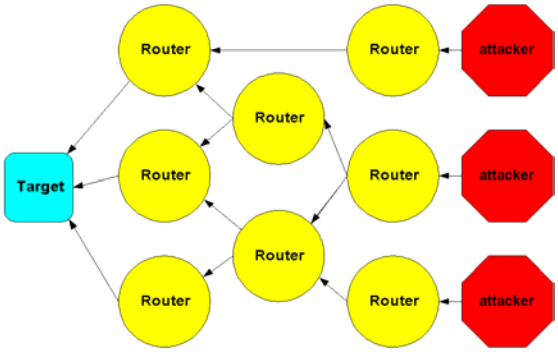
Tele-medicine

- Specialist doctor can perform operations from miles away
- Manipulates robotic arm at local hospital from remote office
- Communication system must be extremely reliable

Need **predictable** ultra-robust real-time man-machine interface technology

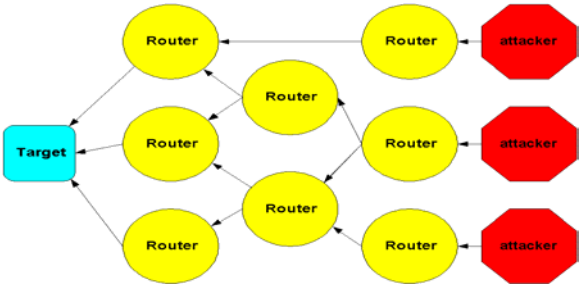
More Internet-dependent!

More susceptible to attacks, e.g., DDOS



Defense against DDOS by "Pushback"

- Upon attack, target launches probes to instruct neighboring routers to recognize attack packets by sampling incoming packets that match attacker's signature
- Routers trace attack packets to immediate predecessor routers from which the attack packets arrive and then launch probes to instruct those predecessors to recognize attack packets, and this repeats upstream
- Routers at the edge of the network shut off attack traffic at the sources
- This is a **real-time contest** between attacker and defender



Far Into the Future

If AIBO were smart enough, she would have asked:

"Why should I roll over on your command?"

Introspective real-time agents need *free will*

The Ultimate Real-Time Systems Research Problem

Cognitive Architecture of Introspective Real-Time Agents

"Men believe themselves free, because they are conscious of their volitions and their appetite, and do not think, even in their dreams, of the causes by which they are disposed to wanting and willing, because they are ignorant of [those causes]."

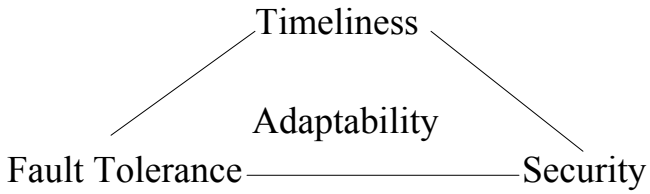
Baruah Spinoza

The Ethics, book 1

Free will is a computational mechanism that prevents an introspective real-time agent from becoming ensnared in an infinite loop.

What cognitive architecture can implement *free will*?

You are the masters of the 21st century technology



- Real-time computing technology is a cross-cutting, enabling technology for all national economies
- We need all of you to help provide solutions to the real-time system challenges of the 21st century

Some Resources and Reading Material

- IEEE Technical Committee on Real-Time Systems
  http://cs-www.bu.edu/pub/ieee-rts/Home.html
- Proceedings of the IEEE Real-Time Systems Symposium (RTSS 25th Anniversary in 2004)
- Proceedings of the IEEE Real-Time Technology and Applications Symposium (RTAS)
- Proceedings of the Euromicro Conference on Real-Time Systems (ECRTS)
- "Real Time Systems", by Jane Liu, Prentice Hall, 2000
- "Real-Time Systems and Software", by Alan Shaw, John Wiley & Sons, 2001