



E. Douglas Jensen's
Real-Time for the Real World

[Home](#) | [Search](#) | [Contact Me](#)

My personal manifesto about the widely misunderstood field of real-time computing...

Navigation

[Introduction](#)
[About Me](#)
[Real-Time](#)
[Distributed Real-Time](#)
[Distrib. Real-Time Java](#)
[Real-Time Java](#)
[Real-Time CORBA](#)
[Real-Time Resources](#)
[Our Documents](#)

Hard and Soft Real-Time

An understanding of time constraints (especially as expressed with time/utility functions), and of the two timeliness factors in a sequencing criterion (the optimality of timeliness and the predictability of that optimality), leads to more general and precise definitions of hard and soft real-time than are usual in the real-time computing field. These enable greatly improved perception of, and cost-effective satisfaction of, system timeliness requirements.

Recall from the [Time Constraints](#) page, in our timeliness model, a hard deadline is not necessarily one that cannot be missed - that is a widespread confusion between the purpose of a time constraint and the purpose of a sequencing timeliness criterion. Rather, a hard deadline simply specifies that the action is untimely if the deadline is missed - it deliberately does not specify the (system-, application-, and situation-specific) consequences of that untimely action. Those consequences are a principal basis for specifying the sequencing timeliness criterion.

A *hard real-time* system is one whose sequencing timeliness factors (there also may be non-timeliness factors) are

- optimality is the binary case that meeting all hard

Last updated: 02/29/2004 17:34:33

News

Real-Time

[Real-Time Overview](#)

[Time Constraints](#)

[Deadlines](#)

[Time/Utility Functions](#)

[Time Constraints Scopes and Priorities](#)

[Sequencing](#)

[Sequencing Criteria](#)

[Timeliness Optimality](#)

[Predictability](#)

[Hard and Soft Real-Time](#)

[Sequencing Algorithms](#)

[Worked Examples](#)

[Coastal Air Defense](#)

- deadlines is optimal and otherwise is suboptimal (in some system-, application-, or situation-specific way)
- predictability of optimality is deterministic.

[AWACS Tracker](#)

[History](#)

These are the only hard real-time sequencing timeliness factors, and a hard real-time system has only these timeliness factors in its sequencing criterion. In the sequencing timeliness criterion 2-dimensional space of optimality and predictability of optimality, hard real-time is at the maximum optimality/maximum predictability corner point; the two factors are not traded off. This definition corresponds to the consensus within the real-time computing research community that hard real-time means all hard deadlines are always met. (The real-time computing practitioner (user, vendor) community has no consensus on the meanings of "hard real-time" and "soft real-time," they use these terms in many different ill-defined ways.)

It is relatively unusual for a computing system to be intrinsically hard real-time. Most non-trivial real-time systems have execution entities with a mixture of hard deadlines and softer time constraints, such as deadlines and the equivalent of time/utility functions (although not usually understood and expressed that explicitly), plus execution entities that have no time constraints (but are subject to non-timeliness sequencing factors). Hard real-time systems typically arise as follows:

- either all the execution entities having hard deadlines are gathered together to form a hard real-time system (usually a "front end" subsystem at the device level) paired with a soft real-time, or (usually) non-real-time, "back end" system (for human interface, database, etc.), because
 - the soft/non-real-time system is a given and can't support hard deadlines
 - or the system designers/implementers don't know how to accommodate mixed time constraints in a single system
- or all the time constraints are artificially forced to be

hard deadlines because the system, or its designers/implementers/users, can't deal with any other kind of time constraints.

Partitioning a computing system into a hard real-time "front end" and a non-real-time "back end" can be a natural and effective approach in some cases. But in many other cases it limits the effectiveness of front end computing, by restricting it to hard deadlines and the hard real-time sequencing timeliness factors, or of back end computing by preventing it from employing time-constraint driven resource management.

Forcing all time constraints to be hard deadlines often limits the system's flexibility and adaptability, while increasing the hardware resource requirements and lowering the hardware resource efficiency.

It is possible to build computing systems that can concurrently support actions having mixtures of hard deadlines, deadlines, soft time constraints, and no time constraints. Usually deadline-based sequencing approaches (in the real-time computing research literature - such approaches are uncommon in practice, as has been explained earlier) allow actions without hard deadlines to be sequenced in any manner that does not interfere with meeting the hard deadlines; only a small fraction of these approaches are secondarily concerned with the timeliness of actions without hard deadlines.

A *soft real-time system* is one that is not the hard real-time special case, and is thus the general case. The sequencing timeliness optimality factor may be anything - examples of factors used very widely (outside the traditional real-time computing community) are "minimize the number of missed deadlines," and "minimize total tardiness." The predictability of optimality is non-deterministic, and is often modeled stochastically. Very common examples (outside the traditional real-time computing community) of sequencing timeliness criteria in terms of both factors are "minimize the expected number of missed deadlines," and "minimize the mean total

tardiness."

In the sequencing timeliness criterion 2-dimensional space of optimality and predictability of optimality, soft real-time is the entire space except for the hard real-time corner point. A system can be considered to be a hard real-time one to the degree that it has hard deadlines and a sequencing timeliness factor that includes always meeting all the hard deadlines.

Some soft real-time systems are non-stochastically non-deterministic. They have properties that are so asynchronous - in the sense of intermittent, irregular, either interdependent or competitive - that stochastic models of predictability for them are either unknown or computationally intractable. Reasoning about the sequence timeliness of such systems is typically performed using simulation models or extensional (rule-based) and other models from fields such as artificial intelligence, decision theory, etc.

Hard real-time systems must have at least some actions with hard deadlines. But the converse is not true - soft real-time systems may have actions with hard deadlines. Those systems are soft real-time because they do not employ the hard real-time sequencing timeliness criterion, they employ soft real-time ones such as "minimize the expected number of missed deadlines," regardless of whether the deadlines are hard or soft.

It is clear that in the technical sense defined here (as opposed to popular misuse by practitioners), soft real-time systems are considerably more difficult to create than are hard real-time ones. Some of this disparity is the intrinsically greater complexity of soft real-time applications, systems, and execution environments. But some is only a transient artifact - both theory and practice of real-time computing systems have historically focused primarily on hard real-time, and that is necessarily changing.

Note that "hard real-time" and "soft real-time" apply only to a

"system," because their definitions are based on sequencing. In this sense, a *system* is any resource management facility that includes sequencing (of threads, only by our convention in this document), anywhere from the hardware up through the OS and middleware to the application(s).

[To be continued...]

Next: [Sequencing Algorithms and Heuristics](#)

Back to: [Time Constraints](#) or [Time/Utility Functions](#) or [Sequencing](#)



[Add to Favorites](#)



[Print Page](#)



[Download a PDF copy of this page](#)

[View Site Changes](#) | [XML](#) | [Site Updated 02/29/2004](#) | [Legal](#)