Adaptive Time-Critical Resource Management Using Time/Utility Functions: Past, Present, and Future

Peng Li, Binoy Ravindran
ECE Department, Virginia Tech
Blacksburg, VA 24061, USA
{peli2,binoy}@vt.edu

E. Douglas Jensen
The MITRE Corporation
Bedford, MA 01730, USA
jensen@mitre.org

Abstract

Time/utility function time constraints (or TUFs) and utility accrual (UA) scheduling optimality criteria, constitute, arguably, the most effective and broadest approach for adaptive, dynamic time-critical resource management. A TUF, which is a generalization of the classical deadline constraint, specifies the utility of completing an application activity as an application- or situation-specific function of that activity's completion time. With TUF time constraints, timeliness optimality criteria can be specified in terms of accrued (e.g., summed) activity utilities. This paper overviews past and recent advances on adaptive resource management for dynamic time-critical systems using UA algorithms. Emerging challenges and new research directions are also identified.

1. Introduction

Time-critical resource management is fundamentally concerned with satisfying application time constraints. The most widely studied time constraint is the deadline. With deadline time constraints, one can specify timeliness optimality criteria such as "meet all deadlines," "minimize maximum lateness," and "minimize deadline-miss rate," and use deadline-driven real-time scheduling algorithms [2] to achieve them.

There are two fundamental problems with deadline-based timeliness optimality criteria. First, the relative utilities of activities in non-trivial real-time systems are often not directly related to their urgency, which is expressed using deadlines. Secondly, the semantics of deadlines become problematic for activities whose attained utilities can *vary* (e.g., increase, decrease) depending on completion time.

Jensen's time/utility functions (abbreviated here as TUFs) [7] generalizes the deadline constraint. A TUF specifies the utility to the system that results from the completion of an activity as a function of the activity's completion time—i.e., the utility is a function of *when* that activity completes. Examples of TUFs are shown in Figure 1.

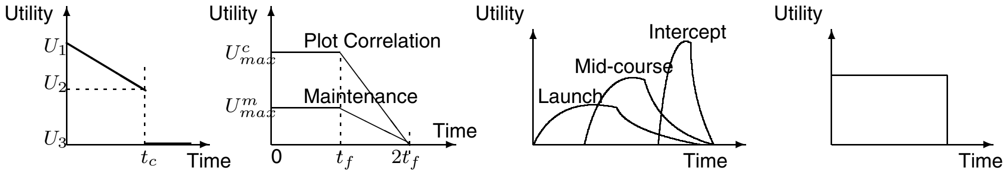
The TUF paradigm has emerged, arguably, as the most effective and most general approach for adaptive resource management in dynamic time-critical systems. In this paper, we survey past research as well as recent advances on TUFs in Sections 2 and 3, respectively. Finally, we identify challenges and new research directions in Section 4.

2 Past TUF Research

To overcome the difficulty with deadlines in radar scheduling problems, Jensen [6] created a more general and expressive model for time constraints – i.e., utility-based scheduling. Although that specific work was, and remains, classified, it was followed on by Jensen's Archons Project at CMU [7] and by Jensen's research and product development projects at subsequent employers and collaborators.

Jensen's Archons project produced two uniprocessor scheduling algorithms [5, 11], which are the first public UA scheduling algorithms. Locke's Best Effort (BE) algorithm [11] can deal with tasks with almost arbitrary TUFs and stochastic parameters, but no resource dependencies. Clark's algorithm, called Dependent Activity Scheduling Algorithm (DASA) can schedule tasks with resource dependencies. However, TUFs in DASA are restricted to downward step functions. Similar algorithms were later implemented and demonstrated in two significant applications, namely the AWACS (Airborne WArning and Control System) surveillance mode tracker system [4] and a coastal air defense system [12].

Furthermore, for restricted tasks with downward step TUFs, Baruah et al. established a theoretical result [1]: the upper bound on the competitive ratio of any on-line algorithm in the presence of overload is $1/(1+\sqrt{k})^2$, where k is the importance ratio of the task set. This upper bound was later achieved by the D^{over} algorithm [8].



(a) Track Asso. (b) Corrln. & Maint. (c) Missile Control (d) A Step TUF

Figure 1. Example Time Constraints Specified Using Time/Utility Functions

3 Recent Advances on TUF Research

Recent interest in TUF research is largely due to the recognition of inevitable overloads in dynamic real-time systems (see [2] for examples of overload scheduling algorithms). However, overload scheduling in real-time computing systems research has traditionally focused on step TUFs, which restricts its applicability in large, complex, dynamic real-time systems. This focus is being changed by algorithms that consider extended task models—e.g., non-step TUFs.

In [13], the authors propose a new packet scheduling algorithm to deal with non-increasing TUFs, which is shown to outperform the previous UA packet scheduling algorithm [3]. In the meanwhile, several process scheduling algorithms have been proposed for variety of task and resource models. The Generic Utility Scheduling (GUS) algorithm [9] is the first algorithm that allows almost arbitrary TUFs as well as resource dependencies. This work was subsequently extended to allow multi-unit resources [14]. Scheduling tasks with non-CPU resource constraints, such as energy, is considered in [15].

Another research area is to schedule end-to-end activities with TUF time constraints. This problem is partially addressed by *Tempus* middleware [10].

Considerable TUF/UA research and technology transition is happening in the classified US Department of Defense (DoD) context.

4 Challenges and New Directions

A major challenge is to provide *assurances* on system behavior, including that on individual and collective timeliness behavior. In [9], the authors develop an algorithm with probabilistic assurance in the context of TUF/UA scheduling. However, there are only a few analytical results for TUF/UA scheduling. A strong analytical foundation, comparable to that of classical deadline-based scheduling theories is thus needed and being sought.

Other challenges include developing a methodology for designing TUFs, which currently is primarily empirically based. Finally, software tool support for TUF/UA scheduling is essential—a proof of concept tool is expected to appear this year.

References

[1] S. Baruah, G. Koren, B. Mishra, A. Raghunathan, L. Rosier, and D. Shasha. On-line scheduling in the presence of overload. In *Proc. IEEE Annual Symposium on Foundations of Computer Science*, pages 101–110, October 1991.

[2] G. C. Buttazzo. *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*. Kluwer Academic Publishers, 1997.

[3] K. Chen and P. Muhlethaler. A scheduling algorithm for tasks described by time value function. *Real-Time Systems*, 10(3):293–312, May 1996.

[4] R. Clark et al. An adaptive, distributed airborne tracking system. In *IEEE WPDRTS*, volume 1586, pages 353–362, April 1999.

[5] R. K. Clark. *Scheduling Dependent Real-Time Activities*. PhD thesis, CMU, 1990.

[6] M. G. Gouda, Y.-W. Han, E. D. Jensen, W. D. Johnson, and R. Y. Kain. *Radar Scheduling: Section 1, The Scheduling Problem*, volume IV of *Distributed Data Processing Technology*, chapter 3. Honeywell Systems and Research Center, 1977.

[7] E. D. Jensen, C. D. Locke, and H. Tokuda. A time-driven scheduling model for real-time systems. In *IEEE RTSS*, pages 112–122, December 1985.

[8] G. Koren and D. Shasha. D-over: An optimal on-line scheduling algorithm for overloaded real-time systems. In *IEEE RTSS*, pages 290–299, December 1992.

[9] P. Li. *Utility Accrual Real-Time Scheduling: Models and Algorithms*. PhD thesis, Virginia Tech, 2004.

[10] P. Li, H. Cho, B. Ravindran, and E. D. Jensen. Scheduling distributable real-time threads in tempus middleware. In *IEEE ICPDS*, pages 187–194, 2004.

[11] C. D. Locke. *Best-Effort Decision Making for Real-Time Scheduling*. PhD thesis, CMU, 1986.

[12] D. P. Maynard et al. An example real-time command, control, and battle management application for alpha. Archons Project TR-88121, CMU, Dec. 1988.

[13] J. Wang and B. Ravindran. Time-utility function-driven switched ethernet: Packet scheduling algorithm, implementation, and feasibility analysis. *IEEE TPDS*, 15(2):119–133, February 2004.

[14] H. Wu, B. Ravindran, E. D. Jensen, and U. Balli. Utility accrual scheduling under arbitrary time/utility functions and multiunit resource constraints. In *IEEE RTCSA*, 2004. To appear.

[15] H. Wu, B. Ravindran, E. D. Jensen, and P. Li. Energy-efficient, utility accrual scheduling under resource constraints for mobile embedded systems. In *IEEE EMSOFT*, 2004. To appear.