Predictably Flexible Real-time Systems - a Scheduling Perspective

Gerhard Fohler
Department of Computer Engineering
Mälardalen University, Sweden
gerhard.fohler@mdh.se

Extended Abstract

Historically, realtime systems have been focussed on providing single, specific solutions to single, specific applications, treating all activities with the same methods, geared towards the most demanding scenarios. While the high cost of such approaches is acceptable for applications with dramatic failure consequences, it is no longer justified in a growing number of new applictions. In these, real-time behavior is demanded only for parts of the systems, few faults can be tolerated. Instead of strict real-time behavior for the entire system, these applications demand "also real-time", or some temporal control.

Two fundamental activation paradigms, time triggered (TT) and event triggered (ET), have been considered as having contradicting assumptions, but providing important attributes each, i.e., determinism and flexibility. The choice of paradigm determines the set of types and constraints on tasks during system design. ET, WITH algorithms such as earliest deadline first or fixed priority, provideS for simple dispatching and flexibility. Adding constraints, however, increases scheduling overhead or requires new, specific schedulability tests which may have to be developed yet. Offline scheduling methods mandated by TT can accommodate many specific constraints and include new ones by adding functions, but at the expense of runtime flexibility, in particular the inability to handle aperiodic and sporadic tasks. Consequently, a designer given an application composed of mixed tasks and constraints has to choose which constraints to focus on in the selection of scheduling algorithm; others have to be accommodated as good as possible.

We will present a method, which overcomes this traditional all-or-nothing approach to provide a combination of both even for parts of the system. Thus, the system can be designed such that appropriate methods can be used for individual parts of the system. The design no longer has to select either one paradigm or the other, with the respective advantages or disadvantages, but can take the best selection independent of the paradigm. Complex constraints are resolved by the offline scheduler, whereas flexibility is introduced via online scheduling - while maintaining the strict TT demands.

Our method allows to set the amount of flexibility at runtime during design in a predictable way by including restrictions on task executions as input to the transformation algorithm, thus providing for predictable flexibility.

We will show how predictable flexility can serve as “temporal interface” to express temporal behavior of an application, independently of the use scheduling algorithm. Providing transformation methods, the temporal behavior of an application can be maintained even various actual scheduling algorithms.