Mechanisms for making gain time available in a hierarchical scheduling framework
Reinder J. Bril

Summary
The analysis for the periodic resource model as described in [1] gives rise to over-provisioning, i.e. the amount of capacity required by an application according to the analysis is larger than actually needed during run-time due to the inherently pessimistic assumptions on which the analysis is based. This assignment aims at making the resulting gain-time available, preferably as soon as possible. The assignment should at least result in (i) definitions of such mechanisms, (ii) simulations to illustrate the effectiveness of the mechanisms, and (iii) an implementation of an appropriate mechanism in a system supporting hierarchical scheduling.

Description
Within SAN, we study reservation-based multi-resource management for real-time networked embedded systems, both from a theoretical perspective, i.e. analytically, as well as a practical perspective, i.e. by actually building such resource management systems as extensions of existing real-time operating systems. This work is done in close co-operation with other universities and industrial partners. Reservation-based resource management gives rise to hierarchical scheduling, and the periodic resource model for compositional guarantees described in [1] is a conceivable theoretical approach. A major advantage of that model is the ability to independently analyze the schedulability of a subsystem. A disadvantage of the approach is the inherent pessimism in the analysis, giving rise to over-provisioning of resources and therefore (additional) gain-time during run-time. This assignment aims at making that gain-time available as early as possible, allowing its provision to soft- and firm real-time tasks. Approaches may optionally use both static (i.e. analytical) as well as dynamic (e.g. monitoring) techniques. To that end, various alternatives need to be explored, defined, and simulated, and an appropriate mechanism must be implemented in a system based on, for example, μC/OS-II, VxWorks, or RTAI/Linux.

The assignment should also result in a white-paper, which may be submitted to a workshop (such as OSPERT) or a conference (such as DATE).

References