Proceedings of IEEE Real-Time Systems Symposium (RTSS), pp. 57-67, Dec. 2004.

Compositional Real-Time Scheduling Framework *

Insik Shin and Insup Lee
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104 USA
{ishin,lee}@cis.upenn.edu

Abstract

Our goal is to develop a compositional real-time scheduling framework so that global (system-level) timing properties can be established by composing independently (specified and) analyzed local (component-level) timing properties. The two essential problems in developing such a framework are (1) to abstract the collective real-time requirements of a component as a single real-time requirement and (2) to compose the component demand abstraction results into the system-level real-time requirement. In our earlier work, we addressed the problems using the Liu and Layland periodic model. In this paper, we address the problems using another well-known model, a bounded-delay resource partition model, as a solution model to the problems. To extend our framework to this model, we develop an exact feasibility condition for a set of bounded-delay tasks over a bounded-delay resource partition. In addition, we present simulation results to evaluate the overheads that the component demand abstraction results incur in terms of utilization increase. We also present new utilization bound results on a bounded-delay resource model.

1 Introduction

Component technology has been widely accepted as a methodology for designing large complex systems through systematic abstraction and composition. Component-based design provides a means for decomposing a system into components, allowing the reduction of a single complex design problem into multiple simpler design problems, and composing components into a system through component interfaces that abstract and hide their internal complexity. Component-based design also facilitates the reuse of components that may have been developed in different environments. A central idea in component-based design is to assemble components into a system without violating the principle of *compositionality* such that properties that have been established at the component level also hold at the system level. To preserve compositionality, the properties at the system level need to abstract the collective properties at the component level.

*This research was supported in part by NSF CCR-9988409, NSF CCR-0086147, NSF CCR-0209024, and ARO DAAD19-01-1-0473.

Real-time systems could benefit from component-based design, only if components can be assembled without violating compositionality on timing properties. When the timing properties of components can be analyzed compositionally, component-based real-time systems allow components to be developed and validated independently and to be assembled together without global validation. In the real-time systems research, however, there has been little attention to the problem of supporting compositionality with timing properties. There has been a growing attention to hierarchical scheduling frameworks where components (applications) form a hierarchy [4, 7, 10, 5, 15, 16, 11, 17, 1]. Many studies [4, 7, 10, 5, 15, 16] introduced methods to analyze the schedulability of a component in a hierarchical scheduling framework, but did not address the issues of synthesizing the timing properties of a component. Recently, a few studies [11, 17, 1] began to address the problem of analyzing the timing properties of components compositionally.

Our primary goal is to develop a compositional real-time scheduling framework where global (system-level) timing properties are established by composing together independently (specified and) analyzed local (component-level) timing properties. To develop such a framework, the following two problems need to be addressed. (1) The *scheduling component abstraction* problem is to analyze the timing property of a component independently. We define this problem as abstracting the collective real-time requirements of a component as a single real-time requirement, called *scheduling interface*. Ideally, the single requirement is satisfied, if and only if, the collective requirements of the component are satisfied. (2) The *scheduling component composition* problem is to compose independently analyzed local timing properties into a global timing property. We define this problem as composing the scheduling interfaces of components as a single real-time requirement. Ideally, the single real-time requirement is satisfied, if and only if, the set of components is satisfied.

In a compositional real-time scheduling framework, the major issue is how to define a *scheduling interface model* in order to specify the collective real-time requirements of a component. In our earlier work [17], we showed that a compositional real-time scheduling framework can be developed by using the Liu and Layland periodic model [12] as a scheduling interface model. We addressed the scheduling component abstraction problem by abstracting a set of periodic tasks under EDF or RM scheduling as a single periodic task. When a component exports its periodic scheduling interface to the system, the system can thus treat the component as a single periodic task. Using the same technique, we addressed the scheduling component composition problem by composing a set of periodic scheduling interfaces under EDF or RM scheduling as single periodic scheduling interface.

In this paper, we consider another scheduling interface model for a compositional real-time scheduling framework. Mok et al. [13] introduced a bounded-delay (resource partition) model $\Phi(\alpha, \Delta)$ to specify a partition of a time-shared resource. To use the bounded-delay model together with the periodic model as scheduling interface models, we should be able to address the problem of abstracting a set of periodic and bounded-delay tasks into a single periodic or bounded-delay task. With the known results on periodic tasks [17], we are able to address the problem of abstracting a set of periodic tasks under EDF or RM scheduling as a single periodic or bounded-delay task. For the problem of abstracting a set of bounded-delay tasks as a single periodic or bounded-delay task, a possible approach is to transform a bounded-delay task into a periodic task and then to use known results on periodic tasks. However, this transformation inherently adds more resource demands. There has been no known results on bounded-delay tasks that can be used to address such an abstraction problem without transforming a bounded-delay task into a periodic task. We thus develop an exact feasibility condition that determines whether or not there exists a scheduling algorithm to schedule a set of bounded-delay tasks over a bounded-delay resource partition. With this new result, we show that a compositional real-time scheduling framework can be developed using the bounded-delay model.

This paper also includes new results on utilization bounds and abstraction overhead evaluations. In our earlier work [17], we proposed a periodic resource model $\Gamma(\Pi, \Theta)$ to specify a periodic behavior of a time-shared resource allocation and presented its utilization bounds under EDF and RM scheduling. There have been no known utilization bounds of a bounded-delay resource partition model. In this paper, we present utilization bounds of a bounded-delay resource under EDF and RM scheduling. For a solution to the scheduling component abstraction problem, we found that the solution requires a more resource utilization than the resource utilizations that are required by the task set of a component. We evaluate the overheads that the solution incurs in terms of utilization increase through simulation.

The rest of this paper is organized as follows: Section 2 provides an overview of our compositional framework, system models, and problem statement. Section 3 briefly reviews a bounded-delay resource model, and Section 4 presents conditions under which the schedulability of a component can be exactly analyzed. Section 5 presents utilization bounds of the bounded-delay resource model. Section 6 addresses the scheduling component abstraction problem and explores the abstraction overheads. Section 7 presents related work. Finally, we conclude in Section 8 with discussion on future research.

2. Compositional Framework and Problem Statement

Our goal is to develop a compositional real-time scheduling framework. In this section, we define a compositional real-time scheduling problem and identify issues that need be addressed by a solution. We also provide our system models and problem statement.

2.1 Compositional Framework Overview

Scheduling is to assign resources according to a scheduling algorithm in order to service workloads. We use the term *scheduling component* to mean the basic unit of scheduling and define a scheduling component C as a triple (W, R, A), where W describes the workloads (of applications) supported in the scheduling component, R is a resource model that describes the resource allocations available to the scheduling component, and A is a scheduling algorithm which describes how the workloads share the resources at all times. A resource R is said to be *dedicated* if it is exclusively available to a single scheduling component, or *shared* otherwise. We describe a *hierarchical scheduling framework*, where scheduling components form a hierarchy and a resource is allocated from a parent component to its child components in the hierarchy.

We define the schedulability of a scheduling component $C(W, R, A)$, after defining some necessary terms. The *resource demand* of a scheduling component $C(W, R, A)$ represents the collective resource requirements that its workload set W requests under its scheduling algorithm A. The *demand bound function* $\texttt{dbf}_A(W, t, i)$ of a component $C(W, R, A)$ calculates the maximum possible resource demands that W requests to satisfy the timing requirements of task i under A within a time interval of length t. The *resource supply* of a resource model R represents the amount of resource allocations that R provides. The *supply bound function* $\mathtt{sbf}_R(t)$ of R calculates the minimum possible resource supplies that R provides during a time interval of length t. A resource model R is said to *satisfy* a resource demand of W under A if $\mathtt{dbf}_A(W,t,i) \leq \mathtt{sbf}_R(t)$ for all task $i \in W$ and for all interval length t. We now define the schedulability of a scheduling component as follows: a scheduling component $C(W,R,A)$ is said to be *schedulable*, if and only if, the minimum resource supply of R can satisfy the maximum resource demand of W under A, i.e.,

$$\forall i \in W \; \forall t \quad \mathtt{dbf}_A(W,t,i) \leq \mathtt{sbf}_R(t). \tag{1}$$

It should be noted that we consider the schedulability condition in Eq. (1) as sufficient and necessary. We believe this is a reasonable way to extend the principle of the traditional schedulability definition. The traditional exact schedulability conditions such as the Liu and Layland's EDF schedulability condition [12] have been developed for a situation where each task will request the maximum (worst-case) resource demand every case (with a constant resource supply), even though there may be a task that actually completes without consuming its maximum resource demand. Following this reasoning, we state the exact schedulability condition under the assumption that a resource provides its minimum (worst-case) resource supply, even though the resource may actually provide more than its minimum in some cases.

We define a *(scheduling) component abstraction* problem as abstracting the collective real-time requirements of a component as a single real-time requirement, called *scheduling interface*, without revealing the internal structure of the component, e.g., the number of tasks and its scheduling algorithm. We formulate the problem as follows: given a workload set W and a scheduling algorithm A such that $C(W,R_D,A)$ is schedulable, where R_D is a dedicated resource, the problem is to find an "optimal" shared resource model R such that a scheduling component $C(W,R,A)$ is schedulable. Here, the solution R is called the scheduling interface of the scheduling component C. The optimality over a resource model can be determined with respect to various criteria such as minimizing resource capacity requirements and minimizing context switch overheads. It is desirable that the resource capacity requirement U_R of a scheduling interface R is equal to the total resource utilization U_W of a workload set W. However, U_R can be larger than U_W. We define a *(scheduling) component abstraction overhead* as $U_R/U_W - 1$ to represent a normalized resource utilization increase.

In a hierarchy of scheduling components, a parent component provides resource allocations to its child components. Once a child component C_1 finds a scheduling interface R_1, it exports the scheduling interface to its parent component. The parent component treats the scheduling interface R_1 as a single workload model T_1. As long as the parent component satisfies the resource requirements imposed by the single workload model T_1, the parent component is able to satisfy the resource demand of a child component C_1. This scheme makes it possible for a parent component to supply resources to its child components without controlling (or even knowing) how the child components schedule resources for their own tasks.

We define a *(scheduling) component composition* problem as combining multiple scheduling interfaces into a single scheduling interface without revealing the information of the multiple scheduling interfaces, e.g., the number of scheduling interfaces and a scheduling algorithm for the multiple interfaces. We formulate the component composition problem as follows: given two scheduling components $C(W_1,R_1,A_1)$ and $C(W_2,R_2,A_2)$ such that a scheduling component $C(W,R_D,A)$ is schedulable, where $W = \{R_1,R_2\}$ and R_D is a dedicated resource, the problem is to find a "optimal" shared resource model R such that a scheduling component $C(W,R,A)$ is schedulable, where $W = \{R_1,R_2\}$. Since we formulate the component abstraction and composition problems the same way, it is desirable that a solution to the component abstraction problem be used to solve the component composition problem.

We define a *compositional real-time scheduling framework* as a hierarchical scheduling framework that supports the scheduling component abstractions and compositions, i.e., supports abstracting the collective real-time requirements of a component as a scheduling interface and composing independently analyzed local timing properties into a global timing property.

2.2 Compositional Framework Models

As a workload model in our framework, we consider a periodic task model $T(p,e)$, where p is a period and e is an execution time requirement $(e \leq p)$. A task utilization U_T is defined as e/p. For a workload set $W = \{T_i\}$, a workload utilization U_W is defined as $\sum_{T_i \in W} U_{T_i}$. Let P_{min} denote the smallest period in the workload set W, i.e., $P_{min} = \min_{T_i \in W}\{p_i\}$. We assume that all tasks in a component are synchronous, i.e., they release their initial jobs at the same time. We also assume that each task is independent and preemptive.

As a scheduling algorithm, we consider the earliest deadline first (EDF) algorithm, which is an optimal dynamic scheduling algorithm [12], and the rate monotonic (RM) algorithm, which is an optimal fixed-priority scheduling algorithm [12].

As a resource model, we consider a time-shared resource model. A resource is said to be *partitioned* if it is available

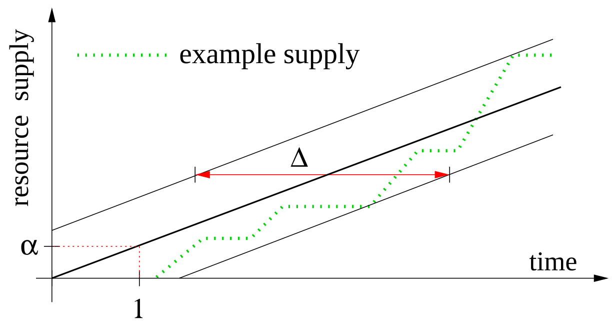

Figure 1. Bounded-delay model: example.

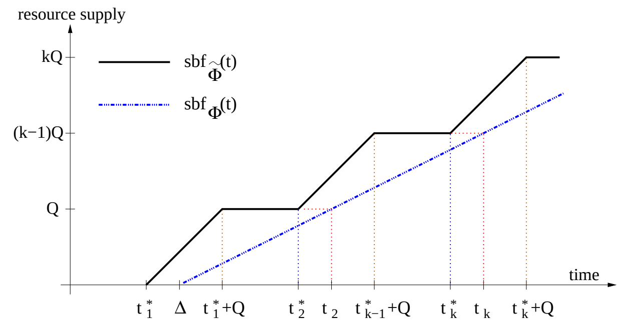

Figure 2. Extended bounded-delay model with scheduling quantum: supply bound function.

to a scheduling component at some times at its full capacity but not available at all at the other times. There are two partitioned resource models: a bounded-delay partitioned model [13] and a periodic resource model [17]. Our goal to investigate how to develop a compositional framework for an arbitrary partitioned model. So far, we have considered these two partitioned resource models: the periodic model in [17] and the bounded-delay model in this paper.

In summary, the problems that we address in this paper are as follows:

- We extend our compositional framework by adding a bounded-delay model $\Phi(\alpha, \Delta)$[13]. For this extension, we develop an exact feasibility condition to determine if a set of bounded-delay workload models is feasible over a bounded-delay resource.
- In our earlier work [17], we presented the utilization bounds of a periodic resource model $\Gamma(\Pi, \Theta)$ under EDF and RM scheduling. In this paper, we develop utilization bounds of a bounded-delay resource $\Phi(\alpha, \Delta)$ under EDF and RM scheduling.
- We evaluate through simulations the overheads that the solution incurs in terms of utilization increase.

3 Bounded-Delay Resource Model

To be able to analyze the schedulability of a scheduling component independent of its context, it is necessary to calculate the resource supply provided to the scheduling component. A resource model is to specify such resource allocations to a scheduling component and to calculate the resource supply to the component. In this section, we briefly review a bounded-delay resource model [13] and provide a new supply bound function for an extended bounded-delay resource model.

Mok et al. [13] introduced a bounded-delay resource partition model $\Phi(\alpha, \Delta)$, where α is an available factor (resource capacity) ($0 < \alpha \leq 1$) and Δ is a partition delay bound ($0 \leq \Delta$). This bounded-delay model $\Phi(\alpha, \Delta)$ is defined to characterize the following property:

$$\forall t_1 \quad \forall t_2 \geq t_1 \quad \forall d \leq \Delta$$
$$(t_2 - t_1 - d)\alpha \leq \text{supply}_\Phi(t_1, t_2) \leq (t_2 - t_1 + d)\alpha.$$

Figure 1 shows a bounded-delay resource example.

For a bounded-delay model Φ, its supply bound function $\mathtt{sbf}_\Phi(t)$ is defined to compute the minimum resource supply for every interval length t as follows:

$$\mathtt{sbf}_\Phi(t) = \begin{cases} \alpha(t - \Delta) & \text{if } (t \geq \Delta), \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

In preemptive scheduling, preemptions may occur at arbitrary time values. However, in discrete-time computing devices, preemptions may only occur at specified discrete intervals. Considering there is a minimum discrete scheduling interval, Feng and Mok [5] introduced an extended bounded-delay model $\tilde{\Phi}(\alpha, \Delta, Q)$, where Q is the minimum scheduling quantum.

The supply bound function of an extended bounded-delay model $\tilde{\Phi}(\alpha, \Delta, Q)$ has not yet been introduced. Thus, we develop its supply bound function $\mathtt{sbf}_{\tilde{\Phi}}(t)$ that computes its minimum resource supply for every interval length t as follows:

$$\mathtt{sbf}_{\tilde{\Phi}}(t) = \begin{cases} t - t_k^* + (k-1) \cdot Q & \text{if } t \in [t_k^*, t_k^* + Q], \\ k \cdot Q & \text{if } t \in [t_k^* + Q, t_{k+1}^*], \end{cases} \tag{3}$$

where

$$t_k^* = t_k - \left\lfloor \frac{t_k}{Q} \right\rfloor Q \tag{4}$$

such that

$$t_k = (k-1)\frac{Q}{\alpha} + \Delta, \quad k = 1, 2, \ldots. \tag{5}$$

Figure 2 illustrates how we define the supply bound function $\mathtt{sbf}_{\tilde{\Phi}}(t)$.

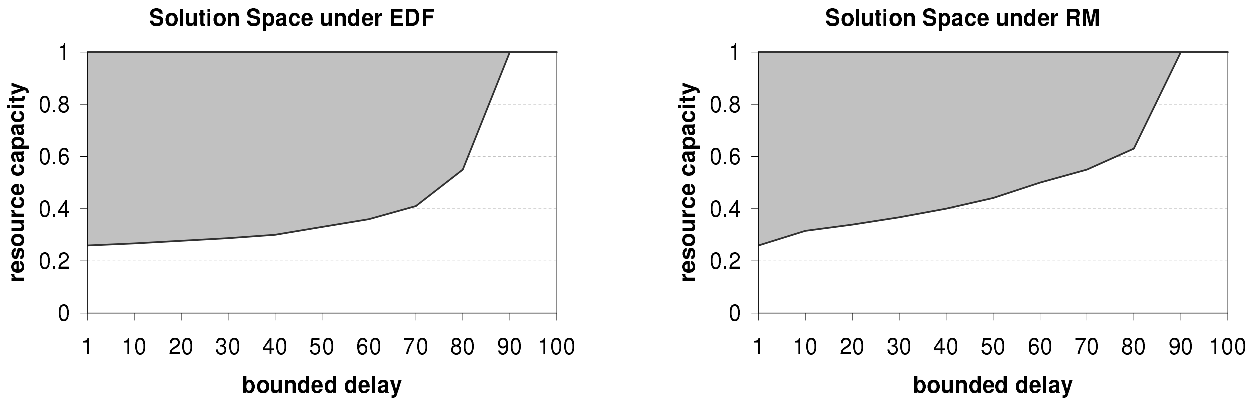

Figure 3. Example of solution space of a bounded-delay scheduling interface model $\Phi(\alpha, \Delta)$ for a workload set $W = \{T_1(100, 11), T_2(150, 22)\}$ under EDF and RM scheduling.

4 Schedulability Analysis

An essential technique to solve the component abstraction and composition problems is to analyze the schedulability of scheduling components. In our earlier work [17], we presented exact conditions under which the schedulability of a scheduling component can be analyzed, when the component consists of a set of periodic workloads and a periodic resource with the EDF or RM scheduling algorithm. In this section, we extend our initial results to include a bounded-delay resource model and address the issues in including a bounded-delay workload model.

4.1 Periodic Workload Model

In our earlier work [17], we developed schedulability conditions for a periodic workload model and a periodic resource model under EDF and RM scheduling. Now, we generalize the schedulability conditions so that it can be used for any partitioned resource model, such as a bounded-delay resource model, as long as the resource model can calculate its supply bound function accurately,

For a periodic task set W under EDF scheduling, Baruah et al. [2] proposed a *demand bound function* that computes the total resource demand $\mathtt{dbf}_{\mathsf{EDF}}(W, t)$ of W for every interval length t:

$$\mathtt{dbf}_{\mathsf{EDF}}(W, t) = \sum_{T_i \in W} \left(\left\lfloor \frac{t - D_i}{p_i} \right\rfloor + 1 \right) \cdot e_i. \qquad (6)$$

We present the following corollary to their result to provide an exact condition under which the schedulability of a component $C(W, R, \mathsf{EDF})$ can be analyzed for any partitioned resource model R.

Corollary 1 *A component $C(W, R, A)$ is schedulable, where $A = \mathsf{EDF}$, if and only if*

$$\forall 0 < t \le 2 \cdot LCM_W + D_{max} \quad \mathtt{dbf}_{\mathsf{EDF}}(W, t) \le \mathtt{sbf}_R(t),$$

where LCM_W is the least common multiple of p_i for all $T_i \in W$ and D_{max} is the maximum relative deadline D_i for all $T_i \in W$.

Proof. This corollary simply follows from Theorem 1 in [17] and can be easily proven by generalizing the proof of Theorem 1 in [17]. □

As an example, let us consider a workload set $W = \{T_1(100, 11), T_2(150, 22)\}$ and a scheduling algorithm $A = \mathsf{EDF}$. The workload utilization U_W is 0.26. With a bounded-delay resource model $\Phi(\alpha, \Delta)$, we now consider a scheduling component $C(W, \Phi, \mathsf{EDF})$. For the problem of guaranteeing the schedulability of the component $C(W, \Phi, \mathsf{EDF})$, a solution space of $\Phi(\alpha, \Delta)$ is shown as the gray area in Figure 3(a). That is, for instance, when $\alpha = 0.4$ and $\Delta = 60$, the scheduling component $C(W, \Phi, \mathsf{EDF})$ is schedulable. We obtain such a solution space of $\Phi(\alpha, \Delta)$ by computing the minimum resource capacity U^*_Φ when the bounded delay Δ is 1, 10, 20, ..., 100, using Corollary 1.

For a periodic task set W under RM scheduling, Lehoczky et al. [8] proposed a demand bound function $\mathtt{dbf}_{\mathsf{RM}}(W, t, i)$ that computes the total resource demand of a task T_i for an interval of length t:

$$\mathtt{dbf}_{\mathsf{RM}}(W, t, i) = e_i + \sum_{T_k \in \mathtt{HP}_W(i)} \left\lceil \frac{t}{p_k} \right\rceil \cdot e_k,$$

where $\mathtt{HP}_W(i)$ is a set of higher-priority tasks than T_i in W.

For a task T_i over a resource model R, the worst-case response time $r_i(R)$ of T_i can be computed as follows:

$$r_i(R) = \min\{t\} \quad \text{such that} \quad \texttt{dbf}_{\mathsf{RM}}(W, t, i) \leq \texttt{sbf}_R(t).$$

We present the following corollary to provide an exact condition under which the schedulability of a component $C(W, R, \mathsf{RM})$ can be analyzed for any partitioned resource model R.

Corollary 2 *A component $C(W, R, A)$ is schedulable, where $A = \mathsf{RM}$, if and only if*

$$\forall T_i \in W \quad \exists 0 < t \leq p_i \quad \texttt{dbf}_{\mathsf{RM}}(W, t, i) \leq \texttt{sbf}_R(t).$$

Proof. This corollary simply follows from Theorem 2 in [17] and can be easily proven by generalizing the proof of Theorem 2 in [17]. □

As an example, we consider a scheduling component $C(W, R, A)$, where $W = \{T_1(100, 11), T_2(150, 22)\}$, $R = \Phi(\alpha, \Delta)$, and $A = \mathsf{RM}$. A solution space of a bounded-delay resource model $\Phi(\alpha, \Delta)$ to guarantee the schedulability of the component $C(W, R, A)$ is shown as the gray area in Figure 3(b). That is, for instance, when $\alpha = 0.4$ and $\Delta = 30$, the component $C(W, R, A)$ is schedulable. We obtain such a solution space of $\Phi(\alpha, \Delta)$ by computing the minimum resource capacity U^*_Φ when the bounded delay Δ is 1, 10, 20, ..., 100, using Corollary 2.

4.2 Bounded-Delay Workload Model

In this paper, we consider a compositional real-time scheduling framework with a bounded-delay model. To develop such a framework with the bounded-delay model, we essentially need to develop a schedulability condition for a bounded-delay workload model as to address the component abstraction problem with the bounded-delay workload model. In this section, we consider the issues of analyzing the schedulability with the bounded-delay workload model and present an exact feasibility condition for a scheduling component that consists of the bounded-delay workload model and a partitioned resource model such as a periodic or bounded-delay resource model.

Feng and Mok [5] presented a condition under which the schedulability of a component can be sufficiently analyzed under EDF and RM scheduling, when the component has a set of bounded-delay workloads and a bounded-delay resource. Their schedulability techniques are to transform each bounded-delay workload model into a periodic workload model and then to analyze the schedulability under EDF and RM scheduling. Following this transformation, we can use Corollary 1 and 2 for the schedulability analysis of a component that consists of a bounded-delay workload model. However, transforming a bounded-delay workload model into a periodic workload model essentially increases a resource demand, which we now explain in detail.

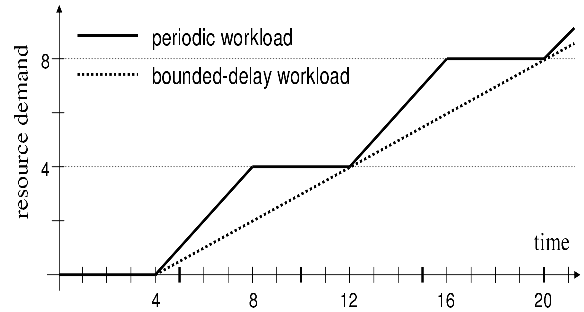

Figure 4. The resource demand compare of a bounded-delay workload $\Phi(\alpha, \Delta)$, where $\alpha = 0.5$ and $\Delta = 4$, and a periodic workload $T(p, e)$, where $p = 8$ and $e = 4$.

Feng and Mok [5] defined a bounded-delay model $\Phi(\alpha, \Delta)$ to represent resource allocations that guarantees at least $\alpha \cdot L$ units of resource allocations in any interval of length $L + \Delta$ for any value of L, i.e., $\texttt{demand}_\Phi(L + \Delta) \geq \alpha \cdot L$. From their definition, we can simply obtain the demand bound function $\texttt{dbf}(\Phi, t)$ of $\Phi(\alpha, \Delta)$ that calculates the minimum acceptable resource demand for an interval length t as follows:

$$\texttt{dbf}(\Phi, t) = \alpha(t - \Delta) \leq \texttt{demand}_\Phi(t). \tag{7}$$

For a periodic workload model $T(p, e)$, its demand bound function $\texttt{dbf}(\Phi, t)$ can be defined as follows:

$$\texttt{dbf}(T, t) = \begin{cases} t - k(p - e) & \text{if } t \in [kp - e, kp], \\ (k - 1)e & \text{otherwise}, \end{cases}$$

where $k = \max(\lceil t/p \rceil, 1)$. To safely transform a bounded-delay workload model $\Phi(\alpha, \Delta)$ into a periodic workload $T(p, e)$ while preserving the resource demand of $\Phi(\alpha, \Delta)$, we need to ensure that $\texttt{dbf}(\Phi, t) \leq \texttt{dbf}(T, t)$ for all t. For instance, one way to safely transform $\Phi(\alpha, \Delta)$ to $T(p, e)$ is given in [13] as follows:

$$p = \frac{\Delta}{1 - \alpha} \qquad e = \alpha \cdot p.$$

Figure 4 shows that the resource demand of a periodic workload model cannot be inherently identical to that of a bounded-delay model, but is supposed be higher to ensure a safe transformation. Thus, we can see that such a safe transformation essentially increases a resource demand.

We now consider the issue of analyzing the schedulability of a scheduling component without transforming a bounded-delay workload model into a periodic workload model. There has been no known scheduling algorithm that can directly handle bounded-delay workloads. Thus, we here consider the problem of analyzing the feasibility of a scheduling component that consists of a bounded-delay workload model. Since we consider discrete-time computing devices, we develop an exact feasibility condition for an extended bounded-delay workload model $\tilde{\Phi}(\alpha, \Delta, Q)$ with scheduling quantum Q.

We note that the demand bound function $\texttt{dbf}(\Phi, t)$ of a bounded-delay resource model Φ in Eq. (7) is equivalent to its supply bound function $\texttt{sbf}_{\Phi}(t)$ in Eq. (2). We apply the same technique that we used in defining the supply bound function $\texttt{sbf}_{\tilde{\Phi}}(t)$ of an extended bounded-delay resource model $\tilde{\Phi}$, to define its demand bound function $\texttt{dbf}(\tilde{\Phi}, t)$. Then, we have the demand bound function $\texttt{dbf}(\tilde{\Phi}, t)$ of an extended bounded-delay resource model $\tilde{\Phi}(\alpha, \Delta, Q)$ as follows:

$$\texttt{dbf}(\tilde{\Phi}, t) = \begin{cases} t - t_k^* + (k-1) \cdot Q & \text{if } t \in [t_k^*, t_k^* + Q], \\ k \cdot Q & \text{if } t \in [t_k^* + Q, t_{k+1}^*], \end{cases} \quad (8)$$

where t_k^* is defined in Eq. (4) and (5).

Now, we present the following theorem to introduce an exact feasibility condition for a component that has a bounded-delay workload set and an extended bounded-delay resource model with scheduling quantum Q.

Theorem 3 *A component $C(W, R, A)$ is feasible, where $W = \{\tilde{\Phi}_i(\alpha_i, \Delta_i, Q)\}, 1 \le i \le n$, and $R = \tilde{\Phi}(\alpha, \Delta, Q)$, if and only if*

$$\forall t > 0 \quad \sum_{i=1}^{n} \texttt{dbf}(\tilde{\Phi}_i, t) \le \texttt{sbf}_{\tilde{\Phi}}(t). \quad (9)$$

Proof. We first consider a real-time job $J(o, e, d)$, where o is an absolute released time, e is an execution time requirement, and d is an absolute deadline. We then construct a mapping from an extended bounded-delay workload $\tilde{\Phi}_i(\alpha_i, \Delta_i, Q)$ to a set of real-time jobs $\{J_{i,k}(o_{i,k}, e_{i,k}, d_{i,k})\}$ such that an individual job $J_{i,k}$ corresponds to the k-th scheduling quantum of $\tilde{\Phi}_i$, $k = 1, 2, \ldots$. That is, when a job $J_{i,k}$ is scheduled, a workload $\tilde{\Phi}_i$ receives its k-th scheduling quantum allocation. Then, we can consider that the job $J_{i,k}$ has a release time of 0, an execution time of Q, and a deadline of $t_{i,k}^* + Q$, where $t_{i,k}^*$ is the latest time instant t such that $\texttt{dbf}(\tilde{\Phi}_i, t) = (k-1)Q$. We define such a mapping systematically as follows:

$$o_{i,k} = 0, \; e_{i,k} = Q, \; \text{and } d_{i,k} = t_{i,k}^* + Q,$$

where

$$t_{i,k}^* = t_{i,k} - \left\lfloor \frac{t_{i,k}}{Q} \right\rfloor Q$$

such that

$$t_{i,k} = (k-1)\frac{Q}{\alpha_i} + \Delta_i, \quad k = 1, 2, \ldots.$$

We consider $W' = \{J_{i,k}\}$, $1 \le i \le n$, $k = 1, 2, \ldots$. Then, the problem of determining whether $C(W, R, A)$ is feasible or not is now equivalent to the problem of determining whether $C(W', R, A')$ is feasible or not.

Consider $A' = \mathsf{EDF}$. The demand bound function of an individual job $J_{i,k}$ can be given as follows:

$$\texttt{dbf}(J_{i,k}, t) = \begin{cases} 0 & \text{if } (t < d_{i,k} - Q), \\ t - (d_{i,k} - Q) & \text{if } (d_{i,k} - Q \le t < d_{i,k}), \\ Q & \text{if } (t \ge d_{i,k}). \end{cases}$$

Then, the demand bound function of a workload set W' under EDF scheduling is simply

$$\texttt{dbf}(W', t) = \sum_{i=1}^{n} \sum_{k=1}^{K_i^*} \texttt{dbf}(J_{i,k}, t),$$

where $K_i^* = \min\{k | d_{i,k} \ge t\}$.

We can simply determine whether $C(W', R, A')$ is schedulable or not, according to Corollary 1.

Finally, we have the following equation by the definition of the mapping from $\tilde{\Phi}_i$ to $\{J_{i,k}\}$:

$$\texttt{dbf}(\tilde{\Phi}_i, t) = \sum_{k=1}^{K_i^*} \texttt{dbf}(J_{i,k}, t),$$

where $K_i^* = \min\{k | d_{i,k} \ge t\}$. □

Example 1 *Let us consider a workload set $W = \{\Phi_1(1/3, 4), \Phi_2(1/4, 6)\}$ and a bounded-delay resource $\Phi(7/12, 4)$. According to Theorem 3, this example component $C(W, \Phi, A)$ is feasible.*

One can see that Theorem 3 is applicable to any resource model R, if the resource model R can calculate its supply bound function accurately, such as a periodic resource model.

5 Utilization Bounds

In this section, we consider a schedulable utilization bound of partitioned resource models. This utilization bound is particularly suited for on-line acceptance tests. When checking whether a new periodic task can be scheduled with existing tasks, computing the utilization bound takes a constant amount of time, much less than the time required to do an exact schedulability analysis based on a demand bound function. In our earlier work [17], we introduced the utilization bounds of a periodic resource model $\Gamma(\Pi, \Theta)$ under EDF and RM scheduling. However, there has been no known utilization bounds of a bounded-delay resource model $\Phi(\alpha, \Delta)$. In this section, we introduce utilization bounds for a bounded-delay resource model $\Phi(\alpha, \Delta)$.

We note that Mok and Feng [14] presented utilization bounds of a partitioned resource that is characterized by a tuple (α, k), where α is a capacity and k is a temporal irregularity[1]. For instance, they provided the following theorem for an EDF utilization bound of a partitioned resource specified by (α, k).

> [Theorem 6 in [14]] A component $C(W, R, A)$ is schedulable, where $W = \{T_i(p_i, e_i)\}$, $A =$ EDF, and R is a partitioned resource with the capacity of α and the *temporal irregularity* of k, if
>
> $$\sum_{T_i \in W} \frac{e_i}{p_i - k} \leq \alpha.$$

We note that the utilization bounds presented in [14], including the above one, are not for a bounded-delay resource model $\Phi(\alpha, \Delta)$, since the temporal irregularity k is not equal to a partition delay bound Δ. In this paper, we derive utilization bounds of a bounded-delay resource model $\Phi(\alpha, \Delta)$ under EDF and RM scheduling.

We present the following theorem to introduce a utilization bound of a bounded-delay resource model $\Phi(\alpha, \Delta)$ for a set of periodic tasks under EDF scheduling.

Theorem 4 *A component $C(W, R, A)$ is schedulable, where $W = \{T_i(p_i, e_i)\}$, $R = \Phi(\alpha, \Delta)$, and $A =$ EDF, if*

$$U_W \leq \alpha\Big(1 - \frac{\Delta}{P_{min}}\Big), \text{ where } P_{min} = \min_{T_i \in W}\{p_i\}.$$

Proof. Due to the space limit, we refer to [18] for a full proof. □

We present another theorem to introduce a utilization bound of a bounded-delay resource model $\Phi(\alpha, \Delta)$ for a set of periodic tasks under RM scheduling.

Theorem 5 *A component $C(W, R, A)$ is schedulable, where $W = \{T_i(p_i, e_i), \ldots, T_n(p_n, e_n)\}$, $R = \Phi(\alpha, \Delta)$, and $A =$ RM, if*

$$U_W \leq \alpha\Big(n(\sqrt[n]{2} - 1) - \frac{\Delta}{2^{(n-1)/n} \cdot P_{min}}\Big),$$

where $P_{min} = \min_{T_i \in W}\{p_i\}$.

[1] We refer interested readers to [14] for the definition of the temporal irregularity of k.

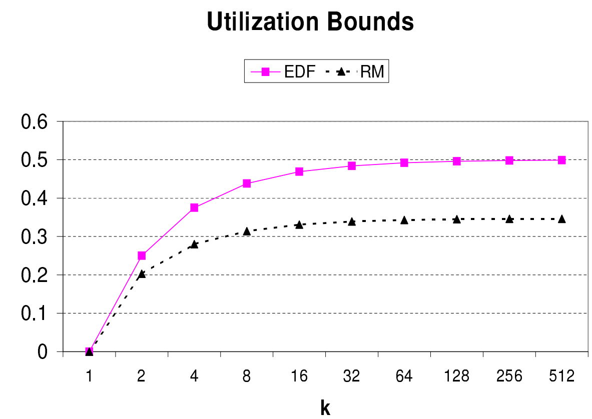

Figure 5. Utilization bounds of a bounded-delay resource model $\Phi(\alpha, \Delta)$, where $\alpha = 0.5$, as a function of k, where $k = P_{min}/\Delta$, under EDF and RM scheduling

Proof. Due to the space limit, we refer to [18] for a full proof. □

Figure 5 shows how the utilization bounds of the bounded-delay resource model grow with respect to k under EDF and RM scheduling, where k represents the relationship between the delay bound Δ and the smallest period in the task set P_{min}, $k = P_{min}/\Delta$. It is shown in the figure that as k increases, the utilization bounds converge to their limits which are α under EDF scheduling in Theorem 4 and $\log 2 \cdot \alpha$ under RM scheduling in Theorem 5.

6 Component Abstraction

We formulated the scheduling component abstraction problem in Section 2 as follows: given a workload set W and a scheduling algorithm A such that a scheduling component $C(W, R_D, A)$ is schedulable, where R_D is a dedicated resource, the problem is to find an "optimal" resource model (scheduling interface model) R such that $C(W, R, A)$ is schedulable. We now illustrate how to address this problem. As an example, let us consider a workload set $W = \{T_1(100, 11), T_2(150, 22)\}$. We consider a bounded-delay model $\Phi(\alpha, \Delta)$ as a scheduling interface model in this example. In addressing the scheduling component abstraction problem for a component $C(W, \Phi, A)$, we can find a solution $\Phi(\alpha, \Delta)$ to this example problem, using Corollary 1 if $A =$ EDF or Corollary 2 if $A =$ RM. In Figure 3, the solution spaces of this example problem are shown as gray areas depending on $A =$ EDF or RM.

In order to derive an "optimal" solution from the solution

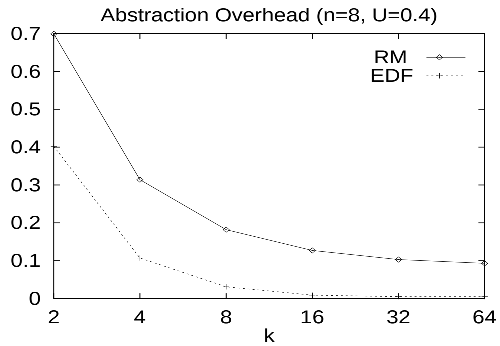

Figure 6. Scheduling component abstraction overheads as a function of k under EDF and RM scheduling, where $k = P_{min}/\Delta$.

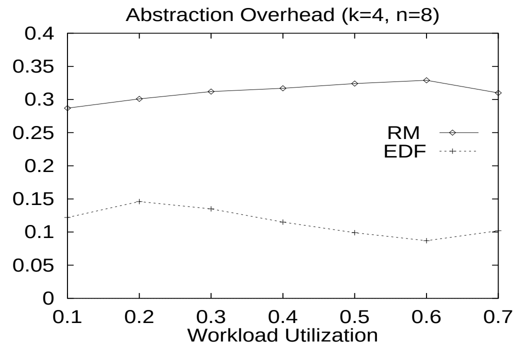

Figure 7. Scheduling component abstraction overheads as a function of workload utilization under EDF and RM scheduling.

space, we now define the optimality criterion as minimizing the resource capacity requirement of a solution when a resource period bound is given.[2] That is, given a workload set W, a scheduling algorithm A, and a bounded delay range such as $\Delta \in [\Delta_{min}, \Delta_{max}]$ such that a scheduling component $C(W, R_D, A)$ is schedulable, the problem is to find a bounded delay resource model $\Phi(\alpha, \Delta)$ such that $C(W, \Phi, A)$ is schedulable and U_Φ is minimized while $\Delta \in [\Delta_{min}, \Delta_{max}]$.

For a scheduling component $C(W, \Phi, A)$, we define its component abstraction overhead as $U_\Phi/U_W - 1$. We performed simulations to evaluate the scheduling and abstraction overheads. For simulation runs, we have used the following settings:

- Workload Size $(|W|)$: The number of tasks in the workload. W is $2, 4, 8, 16, 32, 64$, and 128.
- Workload Utilization (U_W) : The utilization of the workload W is $0.1, 0.2, \ldots, 0.7$.
- Task Model $T(p, e)$: Each task T has a period p randomly generated in the range [5, 100] and an execution time e generated in the range [1, 40].
- Scheduling Algorithm (A) : A is EDF or RM.
- Delay Bound (Δ): The delay bound Δ is determined such that $k = 2, 4, 8, 16, 32$, and 64, where $k = P_{min}/\Delta$ and P_{min} is the smallest task period of a workload set W.

[2] One can extend the optimality criteria by considering some practical issues such as minimizing context switch overheads. However, in this paper, we do not consider such additional issues to focus on the main point of our framework concisely.

Each point shown in Figure 6, 7, and 8 represents the mean of 500 simulation results unless specified otherwise. The 95% confidence intervals for data are within 1-5% of the means shown in the graphs.

Figure 6 plots the component abstraction overheads as a function of k, which represents the relationship between Δ and P_{min}, i.e., $k = P_{min}/\Delta$, where $n = 8$ and $U_W = 0.4$. We can see that the component abstraction overheads significantly depend on k.

Figure 7 plots the component abstraction overheads as a function of workload utilization, where $k = 4$ and $n = 8$. It is shown in the figure that the abstraction overhead is lower under EDF scheduling than under RM scheduling. We can see that the workload utilization is not a significant factor to affect the abstraction overheads.

We also evaluate the component abstraction overheads with respect to the number of tasks n. Figure 8 shows the component abstraction overheads as a function of the number of tasks, where $k = 4$ and $U_W = 0.4$. As stated earlier, each point in the graph is the result of 500 simulation runs, except we performed 200 simulation runs for the case of $n = 128$ under EDF scheduling. We can see that the component abstraction overheads do not increase as the number of tasks increases, but begin to decrease at some point.

The implications of our simulation results are that for a bounded-delay scheduling interface model, its bounded delay is most critical factor for component abstraction overheads, and the workload utilization does not have a relatively considerable impact on the overheads. In addition, we find that the bounded-delay model is a scalable scheduling interface model in terms of component abstraction overhead.

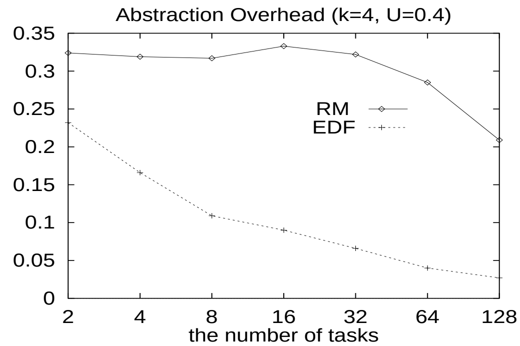

Figure 8. Scheduling component abstraction overheads as a function of workload size under EDF and RM scheduling.

7 Related Work

In the real-time systems research, there is a growing attention to hierarchical scheduling frameworks [4, 7, 10, 5, 15, 16, 11, 17, 1] that support hierarchical resource sharing under different scheduling algorithms.

Deng and Liu [4] introduced a two-level hierarchical scheduling framework where each component (application) can have any scheduler to schedule its tasks while the system has only the EDF scheduler to schedule components. For such a framework, Lipari and Baruah [10] presented exact schedulability conditions, assuming the system scheduler has knowledge of the task-level deadlines of each component. Kuo and Li [7] showed that the RM scheduler can be used as the system scheduler, only when all periodic tasks across components are harmonic. None of these study addressed the component demand abstraction problem.

Mok and Feng [13, 14, 5] proposed a partitioned resource model for a hierarchical scheduling framework. Their bounded-delay resource partition model $R_P(U_P, D_P)$ describes a behavior of a partitioned resource with reference to a fractional resource $R_F(U_P)$. Their model can specify the real-time guarantees that a parent component provides to its child components while any scheduler can work in the parent component as well as in the child components. For their framework where a parent component and their child components are cleanly separated, they presented a sufficient schedulability condition. For a case where a child component has a fixed-priority scheduler, Saewong et al. [16] presented a schedulability analysis based on the worst-case response time calculations. These studies did not address the component demand abstraction problem.

Lipari and Bini [11] and Shin and Lee [17] proposed in parallel a periodic resource model for a compositional hierarchical scheduling framework. Their periodic resource model describes a behavior of a periodic resource and calculates its minimum resource allocations. For a hierarchical scheduling framework where each component can have any scheduler, they presented exact schedulability conditions such that a component is schedulable if and only if its maximum resource demand is no greater than the minimum resource supply given to the component.[3] Based on this schedulability analysis, they both considered the problem of composing the collective real-time requirements of a component into a single real-time requirement by their periodic resource model. Almeida and Pedreiras [1] considered an issue of efficiently solving the component abstraction problem with a periodic scheduling interface model. This paper extends these initial studies by clearly defining a compositional scheduling framework, adding another scheduling interface model, and investigating the overheads that a scheduling interface model incurs in terms of utilization increase.

Regehr and Stankovic [15] introduced another hierarchical scheduling framework that considers various kinds of real-time guarantees. Their work focused on converting one kind of guarantee to another kind of guarantee such that whenever the former is satisfied, the latter is satisfied. With their conversion rules, the schedulability of the child component is sufficiently analyzed such that it is schedulable if its parent component provides real-time guarantees that can be converted to the real-time guarantee that the child component demands. They assumed it is given the real-time guarantee which a child component demands and did not consider the problem of deriving the real-time demands from the child component, which we address in this paper.

8 Conclusion

In this paper, we defined the problems to develop a compositional real-time scheduling framework and presented our approaches to the problems. In addition to a periodic model, we showed that a bounded-delay model can be used as a scheduling interface model for a compositional scheduling framework. We believe that a bounded-delay workload model can be useful to model non-periodic real-time workloads, when we have efficient scheduling mechanisms to schedule the bounded-delay workloads. Thus, our future work is to develop a simple scheduling algorithm that can efficiently schedule bounded-delay workloads.

[3] Lipari and Bini presented their schedulability condition as a sufficient condition. However, we consider it as an exact condition based on our notion of schedulability.

In this paper, we consider scheduling interface models for hard real-time component-based systems. Our future work includes extending our framework for soft real-time component-based systems. This raises the issues of developing soft real-time models for component demand abstraction problems. Soft real-time task models such as the (m, k)-firm deadline model [6] and the weakly hard task model [3] can be useful to develop component demand abstraction models for compositional soft real-time scheduling framework. In this paper, we assume that each task is independent. However, tasks may interact with each other through communications and synchronizations. We also consider extending our framework to deal with this issue.

Acknowledgements

We thank Al Mok for introducing the problem to us and Xiang "Alex" Feng for helpful discussions on their bounded-delay resource partition model. We are also grateful to referees for several useful comments.

References

[1] L. Almeida and P. Pedreiras. Scheduling within temporal partitions: response-time analysis and server design. In *Proc. of the Fourth ACM International Conference on Embedded Software*, September 2004.

[2] S. Baruah, A. Mok, and L. Rosier. Preemptively scheduling hard-real-time sporadic tasks on one processor. In *Proc. of IEEE Real-Time Systems Symposium*, pages 182–190, December 1990.

[3] G. Bernat, A. Burns, and A Llamosi. Weakly hard real-time systems. *IEEE Transactions on Computers*, 50(4):308–321, 2001.

[4] Z. Deng and J. W.-S. Liu. Scheduling real-time applications in an open environment. In *Proc. of IEEE Real-Time Systems Symposium*, pages 308–319, December 1997.

[5] X. Feng and A. Mok. A model of hierarchical real-time virtual resources. In *Proc. of IEEE Real-Time Systems Symposium*, pages 26–35, December 2002.

[6] M. Hamdaoui and P. Ramanathan. A dynamic priority assignment technique for streams with (m, k)-firm deadlines. *IEEE Transactions on Computers*, 44(12):1443–1451, 1995.

[7] T.-W. Kuo and C.H. Li. A fixed-priority-driven open environment for real-time applications. In *Proc. of IEEE Real-Time Systems Symposium*, pages 256–267, December 1999.

[8] J. Lehoczky, L. Sha, and J.K. Strosnider. Enhanced aperiodic responsiveness in hard-real-time environments. In *Proc. of IEEE Real-Time Systems Symposium*, pages 261–270, 1987.

[9] J.Y.T. Leung and J. Whitehead. On the complexity of fixed-priority scheduling of periodic real-time tasks. *Performance Evaluation*, 2:37–250, 1982.

[10] G. Lipari and S. Baruah. A hierarchical extension to the constant bandwidth server framework. In *Proc. of IEEE Real-Time Technology and Applications Symposium*, pages 26–35, May 2001.

[11] G. Lipari and E. Bini. Resource partitioning among real-time applications. In *Proc. of Euromicro Conference on Real-Time Systems*, July 2003.

[12] C.L. Liu and J.W. Layland. Scheduling algorithms for multi-programming in a hard-real-time environment. *Journal of the ACM*, 20(1):46 – 61, 1973.

[13] A. Mok, X. Feng, and D. Chen. Resource partition for real-time systems. In *Proc. of IEEE Real-Time Technology and Applications Symposium*, pages 75–84, May 2001.

[14] A.K. Mok and X. Feng. Towards compositionality in real-time resource partitioning based on regularity bounds. In *Proc. of IEEE Real-Time Systems Symposium*, pages 129–138, December 2001.

[15] J. Regehr and J. Stankovic. HLS: A framework for composing soft real-time schedulers. In *Proc. of IEEE Real-Time Systems Symposium*, pages 3–14, December 2001.

[16] S. Saewong, R. Rajkumar, J.P. Lehoczky, and M.H. Klein. Analysis of hierarchical fixed-priority scheduling. In *Proc. of Euromicro Conference on Real-Time Systems*, June 2002.

[17] I. Shin and I. Lee. Periodic resource model for compositional real-time guarantees. In *Proc. of IEEE Real-Time Systems Symposium*, pages 2–14, December 2003.

[18] I. Shin and I. Lee. Compositional Real-Time Scheduling Framework. Technical report, MS-CIS-04-20, University of Pennsylvania, 2004.