Schedule calculation for arbitrary communication models in component-based real-time systems design

Joseph Porter Abhishek Dubey Graham Hemingway Gabor Karsai Janos Sztipanovits
Institute for Software Integrated Systems, Vanderbilt University, Nashville, TN

Abstract

This is the abstract.

1. Introduction

A number of platform technologies aim to support high-confidence distributed control systems design.

1. Give a short general introduction, so we know where the problem and solution fit in the research field.

CPS -¿ High-confidence -¿ distributed control systems -¿ real-time scheduling

2. Introduce the specific problem and explain what is difficult about it.

More specifically ARINC + distributed objects with multiple communication semantics

Avionics -¿ Partitioned execution + real-time tasks within partitions Avionics -¿ Distributed messaging -¿ non-blocking (time-triggered) + blocking messages

How do we schedule that?

3. Briefly describe our proposed solution to the problem, and our specific contributions to the proposed solution.

Easwaran showed how to do Partitions + EDF on a single processor.

We need to be able to do fixed priority scheduling within partitions, time-triggered message scheduling between partitions on different processors, and guarantee that blocking messages between tasks in any partitions will not cause deadline violations.

We propose to extend the hierarchical scheduling framework originally proposed by Shin and Lee, and extended by Easwaran and Lee.

1. Easwaran's model treats statically scheduled partitions as a periodic supply to the tasks within the partitions. 2. Partition offset bounds must respect message dependency requirements. 3. We assume that blocking messages are sent with worst-case latency - the message is sent at the start of the sending task (which blocks), the receiver gets it at the start of its execution time, and runs to completion before sending a response to unblock the sender.

We are extending our constraint-based scheduling tool to handle the case described above.

Cover related work here in brief.

Old Notes:

1.1. Problems

Semantic gap – mixed design paradigms, and multiple communication models.

- Task execution models: Giotto: ARINC-653 - Communication Message Scheduling: TTP/C TTEthernet - to some extent AFDX - Issue: combining them together and solving an integrated problem.

Examples:

1. Distributed tasks and messaging with time- and event-triggered execution models. Consider the time-triggered ethernet communication protocol.

2. ARINC-653 partitioned execution does not fully specify the semantics of intermodule communication. We consider the case where real-time tasks within a partition make blocking calls to other tasks in the system. Assuming that the dependencies are known *a priori*, the task and partition offsets must be calculated to satisfy the communication constraints. This example assumes an AFDX-style communication model.

1.2. Contributions

Unified scheduling tool, including:

1. Integrated task and message scheduling:

   - Blocking dependencies

     Partitioned execution and communication are critical for secure and fault-tolerant real time systems design. However, distribution of computation over a network complicates partition scheduling, especially in the case of blocking calls (e.g. for remote method invocations). One

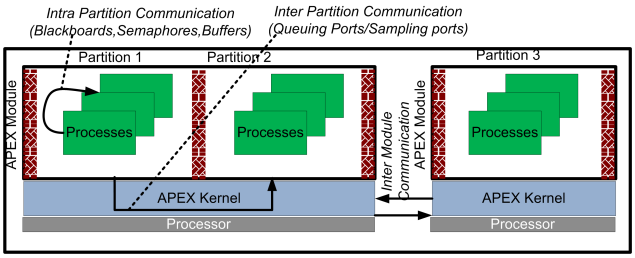

Figure 1. ARINC 653 architecture.

solution to this problem is the use of synchronized partitions and time-triggered data communications. Previous execution models rely on non-blocking communications to enhance fault tolerance, and logical execution time constraints to simplify the calculation of the static message schedule.

We propose to increase the flexibility of this design paradigm by adding blocking calls to partitions. We assume that all partition schedules in the distributed system are synchronized, to ensure proper overlap of blocking messages between partitions. Obviously, the burden of designing the system specifications to eliminate conflicts will fall to the designers. We will also address static *a priori* deadlock detection from the specification.

- Offset determination

2. Two-level scheduling model
    - Partitioned execution
    - Mixed time-triggered and event-triggered approach
3. Flexible communication bus model
4. Extensions of the incremental approach

2. Background

Summaries of everything required to understand the problem and its solution.

ARINC-653 software specification describes the standard Application Executive (APEX) kernel and associated services that are supported by safety-critical real-time operating systems (RTOS) used in avionics. ARINC-653 systems (see figure 1) group *processes* into spatially and temporally separated *partitions*, with one or more partitions assigned to each *module*, and one or more modules (processor hosts) form a system. While spatial partition guarantees total memory isolation between processes in two different partitions, temporal isolation ensures exclusive use of the processing resources by a partition. A fixed priority cyclic schedule is used by the RTOS to share the CPU between partitions. Within each partition, processes are scheduled in a priority preemptive manner.

Processes within a partition share data only via the intra-partition services, and are responsible for their individual state. Intra-partition communication is supported using *buffers* that provide a queue for passing data messages and *blackboards* that allow processes to read, write and clear a single-item data message. Inter-partition communication is asynchronous and is provided using ports and channels that can be used for *sampling* or queuing of messages.

Even though ARINC-653 specification provides a well-defined task execution model, it does not provide enough details about communication schedule. For example, there is no information and support for how a task execution model affects or is dependent upon the order in which the messages are sent over the shared bus.

3. Solution Overview

Details of our particular solution.

4. GPS System Example

Introduce specific example along with details that get all the way down to the measurement points.

5. Acknowledgements

This work is sponsored in part by the National Science Foundation (grant/contract number NSF-CCF-0820088) and by the Air Force Office of Scientific Research, USAF (grant/contract number FA9550-06-0312). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research or the U.S. Government.

References