

**Cost reduction through early schedule
verification and optimization for ARINC
653-based partitioned software systems**

Christoph Ficek
ficek@symtavision.com

Symtavision GmbH
Frankfurter Str. 3 C
38122 Braunschweig
Germany

1 Introduction

In today systems the fixing of errors and failures in late development phases is very expensive. Because of this virtual verification in the design phase is highly desirable. One important aspect for the verification of embedded software is timing and scheduling, especially for safety critical systems.

The ARINC 653 operating system standard facilitates software integration in a protected way (memory and time) according to safety standards DO178 and IEC 61508. Despite its fully deterministic top-level TDMA schedule, there are questions: How to optimize the TDMA layout and maximize utilization? How to verify process deadlines within each partition? Where is the interface between applications development (one partition) and system integration? This requires a more detailed look on timing and scheduling and a reliable scheduling analysis.

From the timing point of view two major challenges exists:

1. Guarantee that each application meets the process deadlines.
2. Optimize the MAF

The first challenge must be done application specific, only knowing the application processes and the MAF. The challenge is to analyze the processes of an application not only considering the possible preemptions of higher priority processes of the same application, but also considering the preemption by other time windows of the MAF. Furthermore dynamic effects like overheads from the operating systems have to be considered:

- Resource access
- Communication drivers (external interrupts from the network)
- Error/exception handling (sporadic occurrence)
- Context switch, between processes / partitions, (replenishment)

The second challenge is a task for the system integrator. He needs knowledge about the applications and must design the MAF such that all deadlines are met and the processor is optimally utilized. For both we essentially need a means to verify process deadlines by discover the worst possible response times of processes. Facing these questions in late phases of the integration leads to problems, which are expensive to solve. Because of the complexity a tool support is essential.

This paper shows how the scheduling for ARINC 653 partitions can be verified in early design phases by *virtual verification*. First the ARINC 653 scheduling strategies are presented and how they are analyzed. Furthermore it is shown how the analysis can be used for verification and optimization of the scheduling and how a design flow looks like for it. Finally, future work and challenges are indicated.

2 ARINC 653

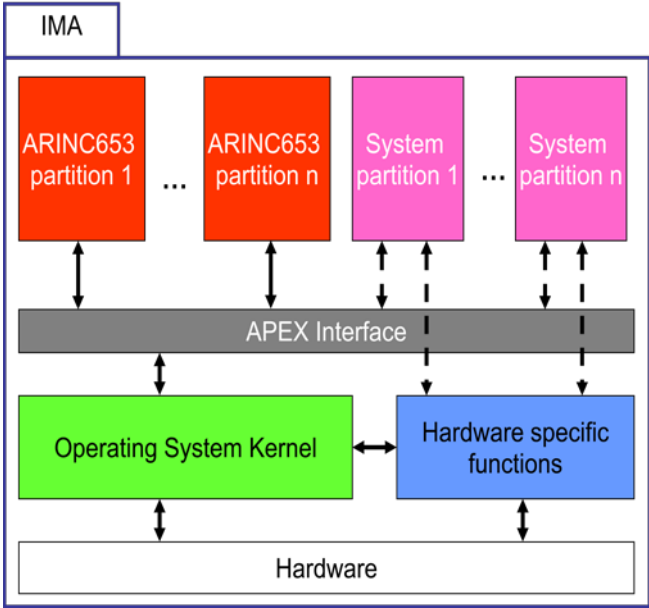


Figure 1: IMA (Integrated Modular Avionics) concept. APEX: Application/Executive

ARINC (Aeronautical Radio inc.) is an US organization which develops standards for many systems in todays aircrafts. One of these standards is ARINC 653 Part 1 – Required Services [2]. The standard defines how a control unit in aircrafts called IMA (Integrated Modular Avionics) works. One of the main aspects is the defined possibility of placing several applications with mixed criticality on the same IMA. The IMA concepts itself assured that in error cases of one application, the other applications are not influenced. This is a high safety aspect of the IMA concept and ARINC 653. This is possible by isolating each application in one partition of the system in their used memory, their I/O possibilities and the processing time. Further ARINC 653 defines an interface called APEX (Application/Executive) which is the only way for a partition to communicate with the operating system (OS). For communication between partitions on the same or different IMAs, defined communication ports exist. Figure 1 shows the IMA concept. The standardized interface between the OS and applications also allows reusing an application in another IMA and changing the operating system to another ARINC 653 conform one. This paper concentrates on the timing aspect of

the IMA concept and ARINC 653. In the following sections the timing relevant mechanisms in ARINC 653 are described.

2.1 IMA concept and Major Frame

The isolation of the partitions in processing time of the CPU is done by TDMA (Time Division Multiple Access). Each partition is granted one or more such TDMA time windows. These TDMA windows form the Major Frame MAF which is repeated periodically and can has idle times, which do not belong to any partition.

The design of the major frame and the TDMA slots is done by the system integrator. He has to map all partitions to one IMA. Each partition has a set of requirements (memory size etc.) which the integrator must fulfill. Two parameters are the partition period and the partition duration. These parameters define how many processing time (duration) must be granted to the partition and in which time intervals (period) these must repeat in the MAF. The duration can be distributed over several windows. The length of the Major Frame, and so its period, is defined in the ARINC 653 standard as an integer multiple of the least common multiple of all partition periods. Figure 2 illustrates this.

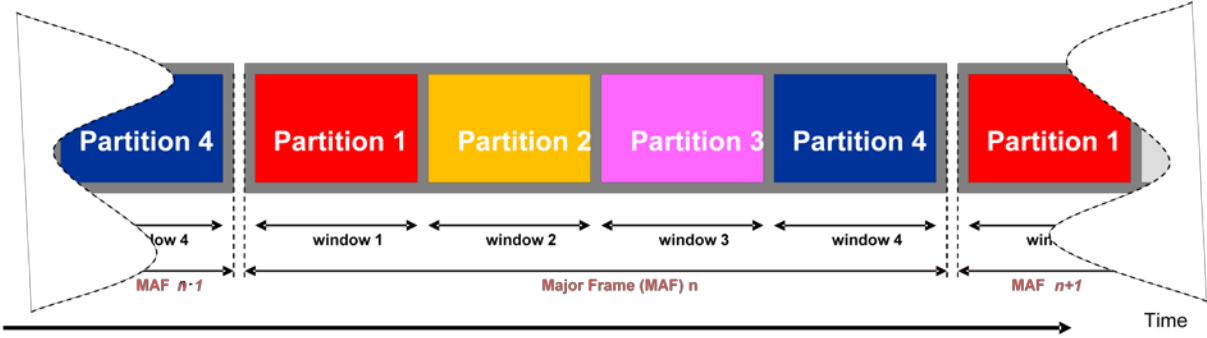


Figure 2: Time window allocation and major frame in ARINC 653

2.2 Intra-Partition schedule

3 Inside the partitions processes of the applications exists. The processes are executed when a time window of the partition is executed. They are scheduled by a Static Priority Preemptive SPP scheduler. The end of a time window will preempt all running processes, they will be resumed in the next time window of their partition.

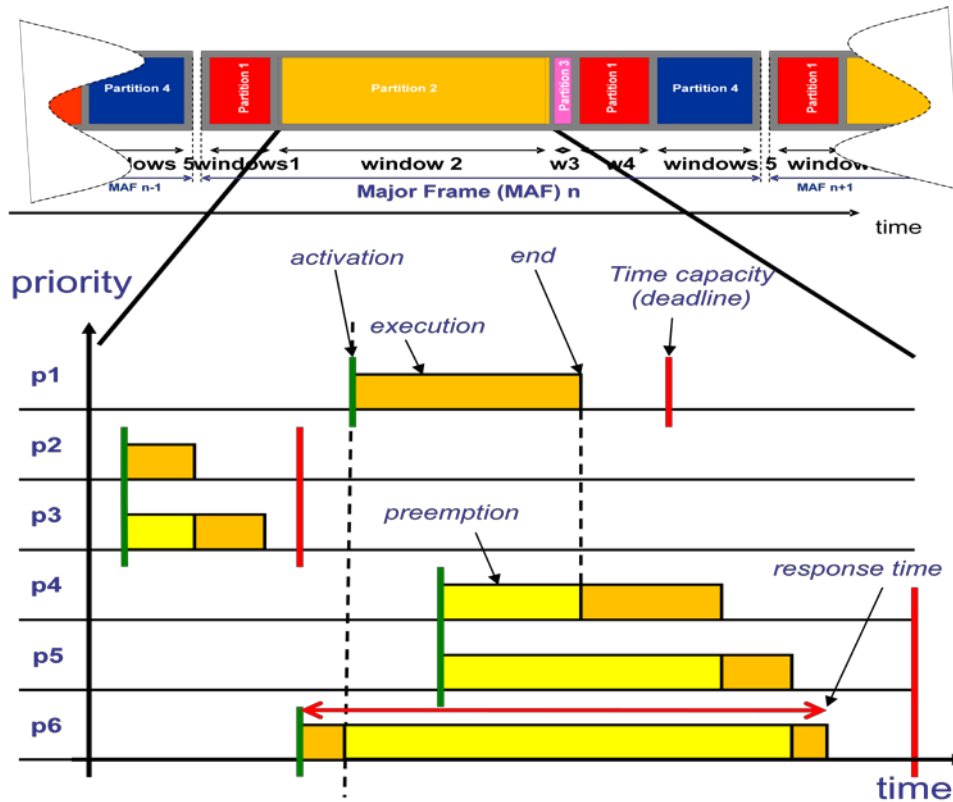


Figure 3: Process execution in one partition.

Figure 3 shows the relationship between the partitions and the processes. It also shows the different process parameters. From a timing point of view, each process is characterized by its priority for the scheduling, the activation pattern (e.g. periodically) and its execution time which it needs on a processor. The response time of a process is the time from its activation until the end of its execution including all times, when the process was preempted. A further parameter called time capacity (deadline) can be specified. The deadline defines the maximum allowed response time for a process. A deadline miss can have several consequences in ARINC 653 depending on the application configuration and the process criticality. In general a deadline miss indicates a failure behavior of a process from a timing point of view and should not occur. An additional parameter for ARINC 653 processes is

called delayed start. This is an offset, which shifts the first release point of a process relative to the first partition window.

3.1 Start time considerations in ARINC 653

The regular scheduling in a partition starts after an initialization phase of the whole IMA module. This phase can be finished at any time after system startup. After the initialization is finished, the ARINC 653 processes of one partition are set to “ready” but (!) do not start execution immediately. Rather, their start is delayed until the first window of the next period of the partition. These windows are marked in ARINC 653 with a parameter partition period start with the values true or false. Because of this, the first activation of an ARINC process and the beginning of a time window of another partition can never arrive at the same time and furthermore the beginning of the scheduling in an IMA is completely deterministic. A side effect is that different starting windows of processes can lead to different worst case scenarios. This has to be considered in the analysis.

4 Virtual verification for ARINC 653

The *virtual verification* starts in the early design phases of systems. It is model based and abstracts in some points from the HW and SW to make fast analysis possible. In this early phase only initial requirements exist often without even the hardware. For a scheduling analysis in these phases budgets or estimations can be used instead of real worst case execution times (WCETs), which are not determined yet. With the budgets the design itself can be proven for scheduability and performance of an application. When the design process makes any kind of iterations, the virtual verification can be made after each.

Changes in the design regarding the timing or the performance (e.g. a function has to be designed bigger (=more WCET)) can be fast added to the model and proven by the analysis. In this way the change can be virtually verified to be feasible or not.

Going further in the design process, at some point real WCETs of the implemented application can be determined by measurements or static code analysis. Also other details like interrupt behavior or OS overheads can add to the model when they are known. In this way the model gets tighter to the reality and the results are also more and more matching the real system in the end.

Using this approach, design faults and late integration problems can be avoided and reduce the cost of late corrections. Furthermore, over dimensioning of systems can be reduced and computation time better exploited by the applications.

A further point is the DO178-B which claims for dealing with the timing of systems. Doing this during the whole design process, grants better confidence in the results and the system behavior.

4.1 Example methodology

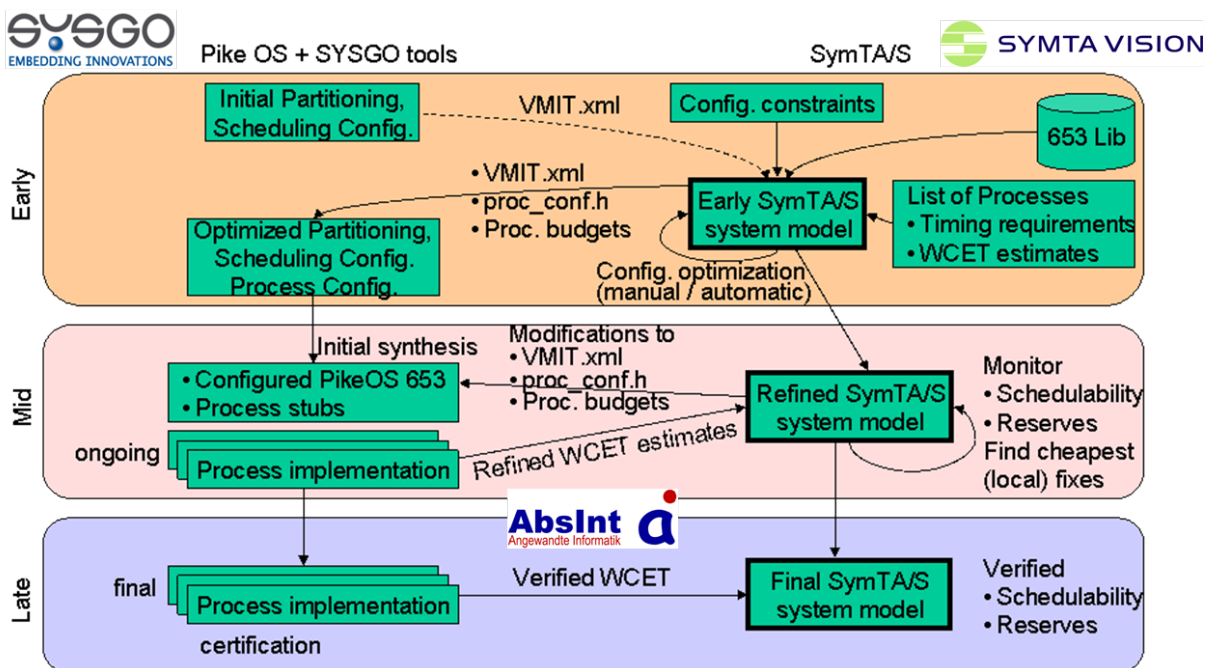


Figure 4: Methodology with Syntavision, Sysgo and AbsInt

Figure 4 shows an example methodology how it was established by Sysgo, AbsInt and Syntavision. Tools by Sysgo was used to design the ARINC 653 system, SymTA/S by Syntavision was used for the scheduling analysis and virtual verification and AbsInt tools for WCET determination using static code analysis. Figure 5 shows a screenshot of SymTA/S. In (1) the project explorer is shown, with the whole model tree of the system. The partition, the processes and window schedules can be found there. (2) shows the parameters of the partition as table and also the load of the partition. The partition duration and the period values are set the same as in the example. (3) shows the table of processes. The system is already analyzed and the response time values for both processes are shown on the right side of the table. (4) is the table of window schedule only of the partition. In (5) a single editor for the major frame MAF is opened and shows the duration of 200ms.

Results of the analysis are the load of each partition and which process is causing how much. Also the worst case Response time (WCRT) as value and Gantt-chart is provided. The Gantt-chart in Figure 5 shows how the WCRT of a specific process could occur and what the worst case is for it.

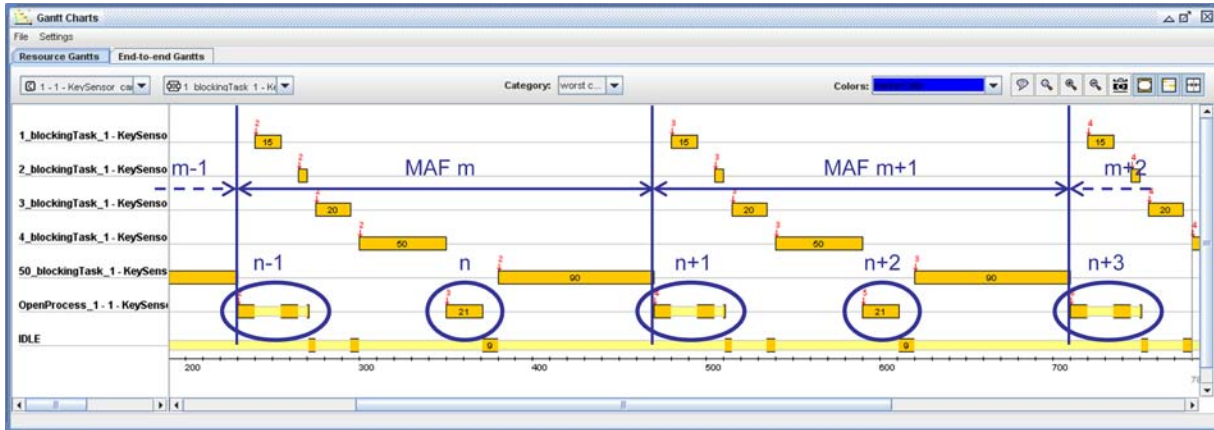


Figure 5: Gantt-Chart example (SymTA/S screenshot)

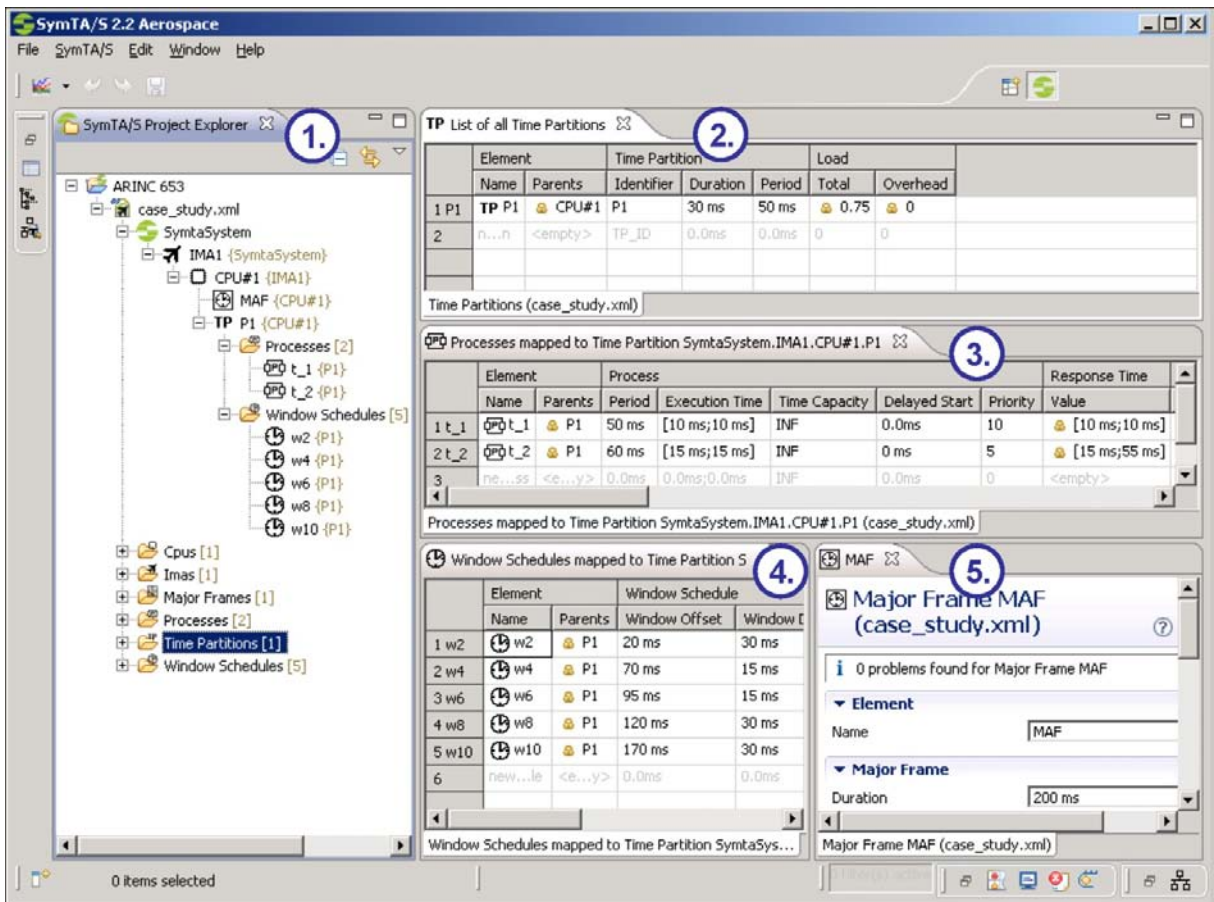


Figure 6: SymTA/S screenshot

5 Conclusion & outlook

It has been shown how the ARINC 653 scheduling works and how complicated it can get. Considering timing and performance issues too late in the design process can lead to high cost. A continuous refinement and proving of the timing is essential to avoid the integration problems. How this can happen has been shown.

Future work will investigate the possibility of automatic Major Frame generation considering all timing effects. Also a coupling of ARINC 653 applications to communication networks like AFDX, CAN or TTP is investigated at the moment. This will allow to consider system level timing including communication data paths.