

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

Christoph Ficek, Symtavision GmbH, Germany  
Avonics & Defence Electronics Europe 2011

Solutions for Complex  
Real-Time Systems



# Symtavisision – Who we are

**Leader in system-level timing-design and verification**

## Company

- ▶ Founded 2005, 25 employees
- ▶ Based in Braunschweig, Germany
  - ▶ Munich branch, distributors in France, Italy, Japan, China, Korea. USA expansion in 2011
- ▶ Markets: Automotive, Aerospace, Automation ...

## Solutions

- ▶ Timing tools: SymTA/S<sup>®</sup>, TraceAnalyzer<sup>™</sup>
- ▶ Standards-based tool-integration
- ▶ Engineering services, methodology consulting



Dr. Jersak  
CEO



Dr. Richter  
CTO



W. Ries  
CSO



Founding member



# Our Customers

## OEMs & Suppliers

- Focus Automotive
- Aerospace, Automation

## Applications

- ▶ Performance- & Safety-Critical Controllers
- ▶ Software-, Network- and System-Architecture
- ▶ System Integration



BMW Group



DAIMLER

VOLVO



DENSO

PSA PEUGEOT CITROËN



FIAT



IVECO



All other products, logos and company names mentioned in this document are trademarks or registered trademarks of their respective companies/owners.

# INTERESTED Aerospace Partners



Project ([www.interested-ip.eu](http://www.interested-ip.eu)), Jan 08 – Apr 11  
Aerospace partners include



THALES



TTEch



AbsInt  
Angewandte Informatik

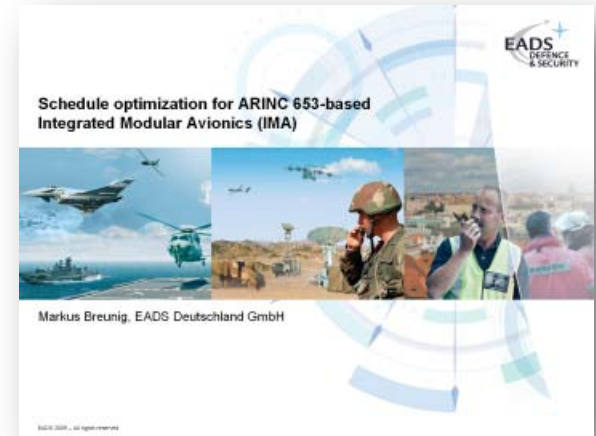


- Results: ARINC-653 (IMA), ARINC-664 (AFDX), TTP scheduling analysis

# Aerospace Customers

## Markus Breunig, EADS D&S:

- With SymTA/S, the usage of common analyzing methods and tools is possible for ARINC 653 scheduling



## Sergio Penna, EMBRAER:

- SymTA/S has proven to be an invaluable tool for analysing a distributed avionics system following the IMA concept



# Partitioned Scheduling in ARINC 653

Solutions for Complex  
Real-Time Systems



# The Challenge

## ❑ System design and virtual timing verification

- ❑ Architecture alternatives
- ❑ Cost effective real-time computation and communication
- ❑ **Avoid late integration problems**
  - **deadline overruns, lost or inconsistent data, excessive jitter ...**



## ❑ Software implementation and refinement

- ❑ Timing requirements and software budgets for suppliers
- ❑ Timing verification for each implementation step
- ❑ Early warning and remedy for impending timing problems



## ❑ System integration and verification of functional safety

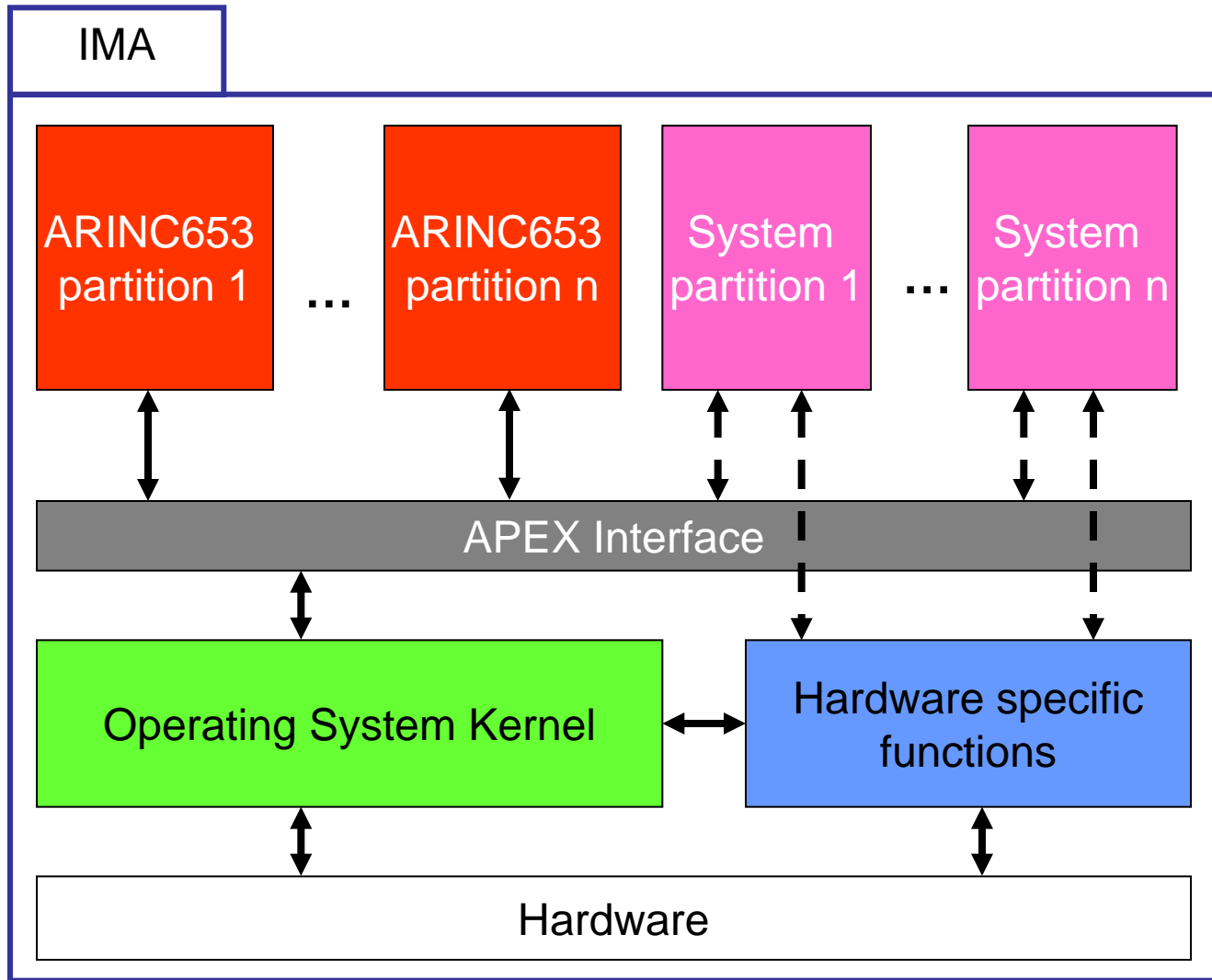
- ❑ Maximize system reliability
- ❑ Functional safety: system is free from unacceptable risk due to timing failures



# Scheduling-Related Questions

- ❑ Do all partitions have enough execution time?
- ❑ Will each process meet its deadline?
- ❑ What is the load of each partition? Is there room for extensions?
- ❑ Is the overall schedule (major frame) balanced? Can it be optimized?
- ❑ Which data is needed for which level of confidence in the analysis?

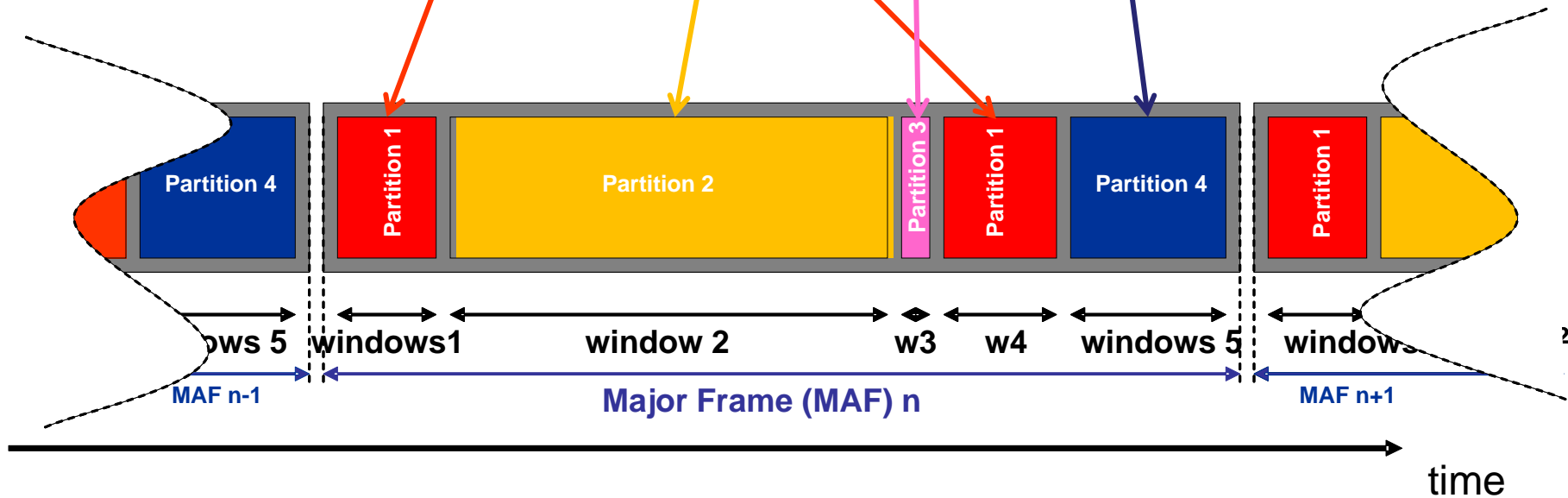
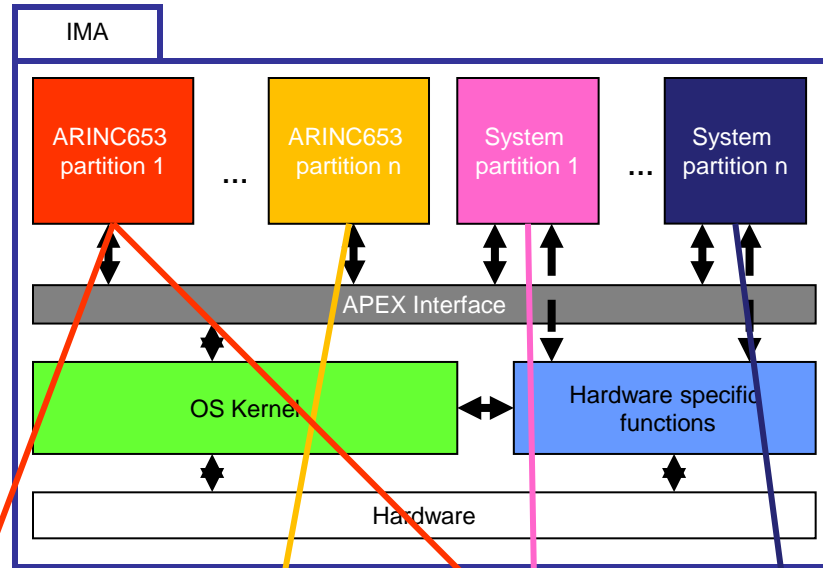
# Partitioned Operating Systems – Virtualization



APEX: Application/Executive

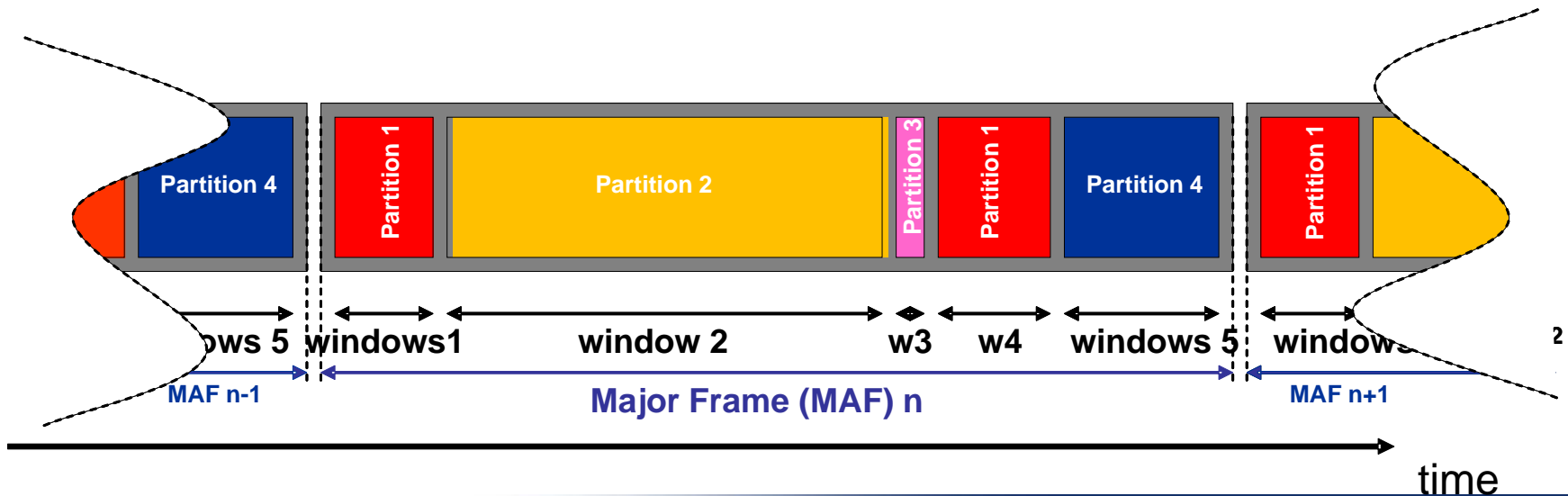
IMA: Integrated Modular Avionics

# Partitioned Scheduling of ARINC 653



# Partitioned Scheduling – Timing Protection

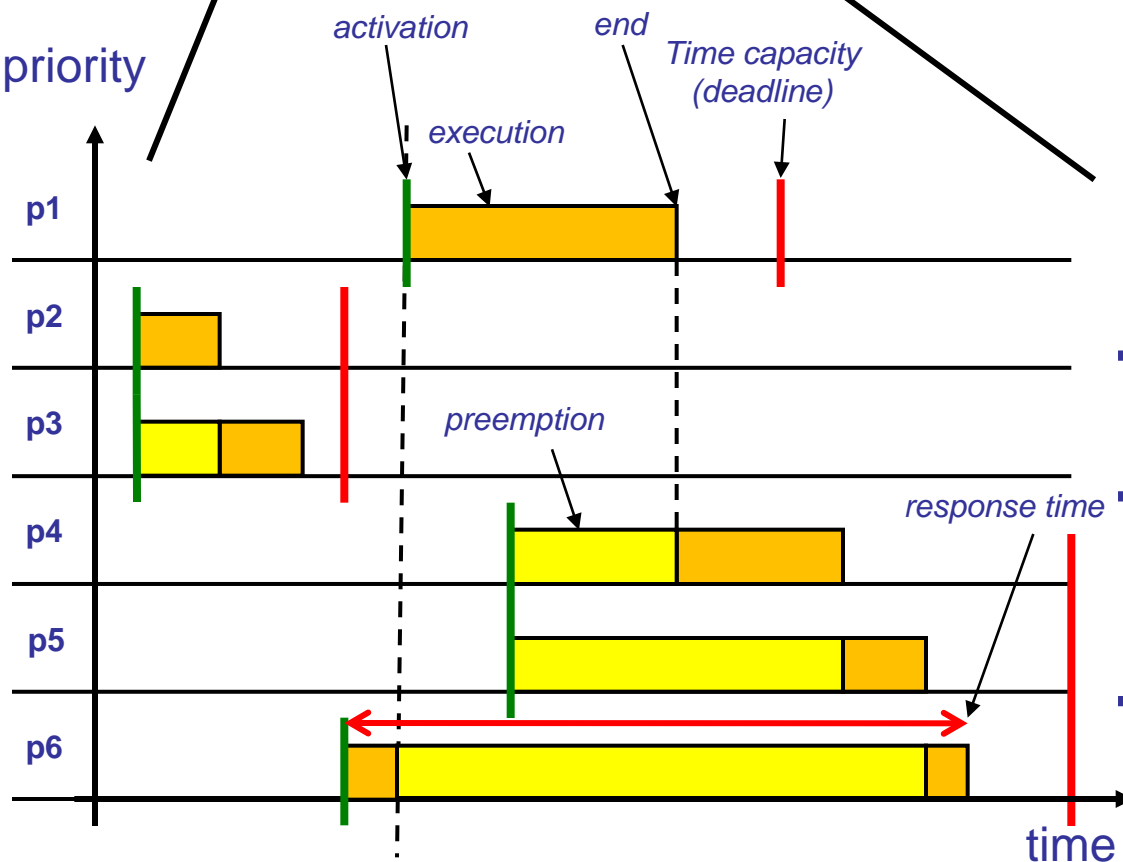
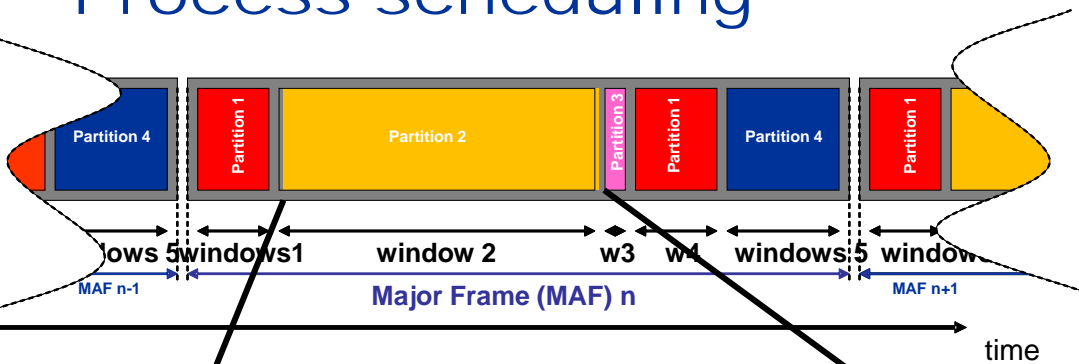
- Static scheduling, Time-Division Multiple Access (TDMA)
  - **guaranteed time-slots** for each partition, no disturbance possible
- Properties of ARINC 653 Scheduling:
  - repetitive execution of **major frames**
  - $m$  **partitions** →  $n$  **sequential windows**



# Major Frames, Windows, Partitions

- ❑ Major Frame has
  - ❑ period
  - ❑ several sequential windows, each has
    - offset
    - duration, i.e. execution time *available*
- ❑ Partitions has
  - ❑ period
  - ❑ duration, i.e. execution time *needed*
- ❑ MAF layout shall match partition requirements
- ➔ partition duration  $<$  sum of window durations  $\times$  period ratio !
  - ❑ can be simple
  - ❑ can be more complex

# Process scheduling



□ Partition software consists of **processes**

- period
- priority (static priority scheduling, SP)
- execution time
- **time capacity (deadline !)**
- ...

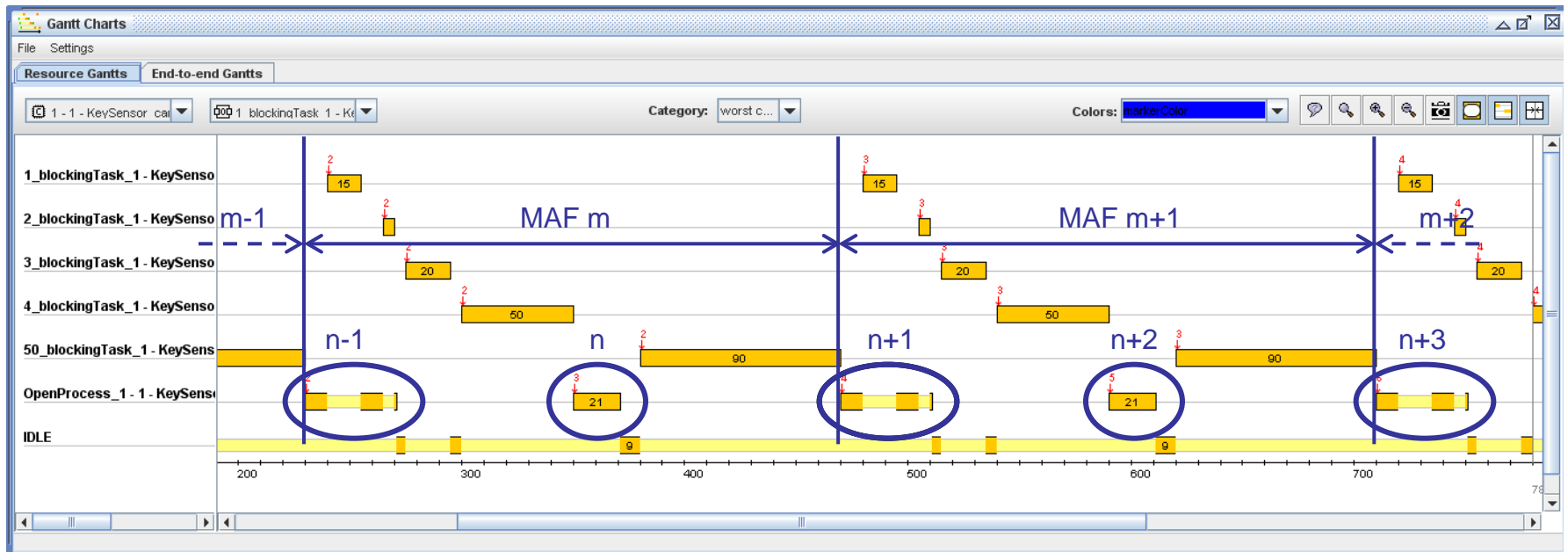
➔ **hierarchical scheduling: SP over TDMA**

➔ **capacity verification requires scheduling analysis**

➔ **must also fit to Major Frame Layout → Optimization**

# Preemptions of Process in Time Partition

- 2 partition executions in 1 MAF, not “symmetric”
- → two different execution timings



- can only be answered with knowledge about MAF and processes

# ARINC 653 Scheduling Analysis

Solutions for Complex  
Real-Time Systems

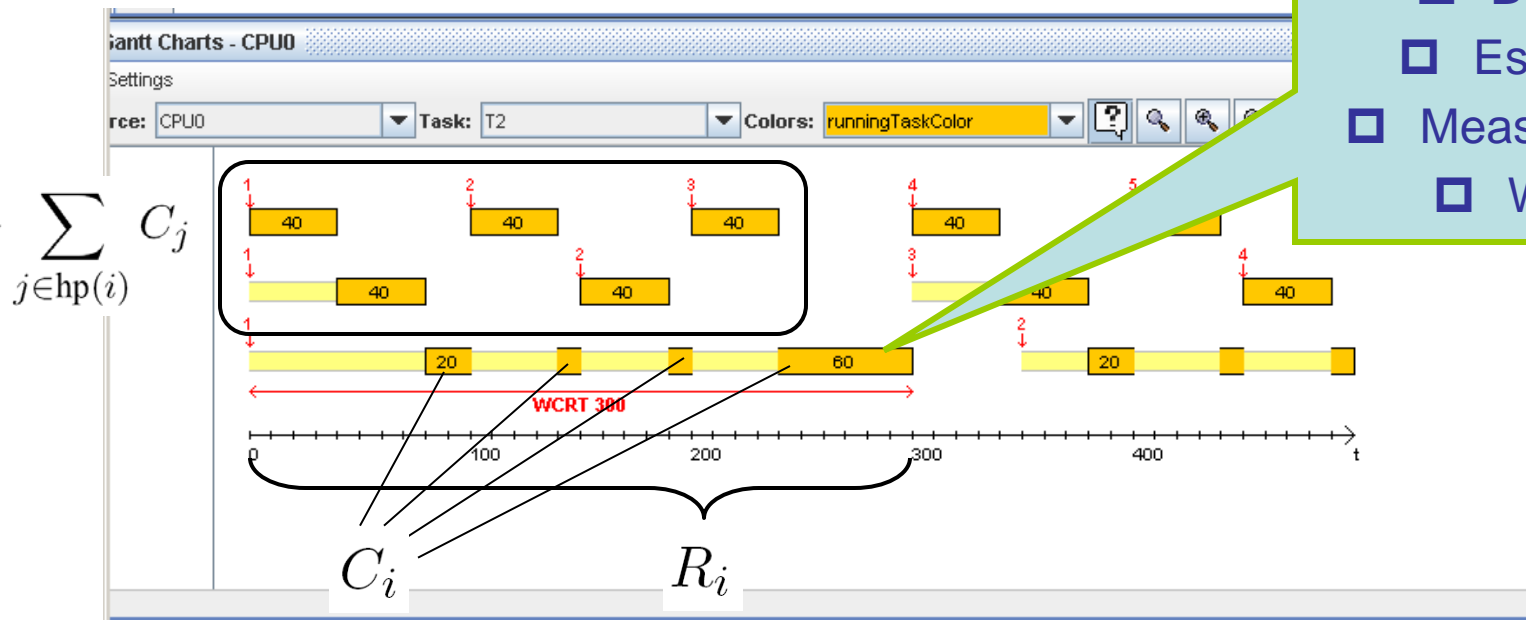


# Scheduling Analysis basics – The response time formula

*fix-point problem*

$$R_i = C_i + \underbrace{\sum_{j \in \text{hp}(i)} C_j \left\lfloor \frac{R_i}{T_j} \right\rfloor}_{\text{interference term } I_i} \leq D_i = T_i$$

↑ response time  
↑ core execution time  
# of preemptions



- $C_i$  can be
- ▣ Budgets
  - ▣ Estimates
  - ▣ Measurements
  - ▣ WCET

# Data needed for ARINC 653 Scheduling Analysis

- ❑ Configuration data is provided by the ARINC 653 standard
  - ❑ Exchange format defined by ARINC 653 XML
  
- ❑ Further parameters needed for scheduling analysis:
  - ❑ Worst case execution time for every process of an analyzed partition
  
- ❑ Amount of execution times depends on focus:
  - ❑ single partition analysis
  - ❑ overall system-level analysis

# Partition period start – processes first release point

- ❑ After the power up of an IMA or a reset all partitions access an initialization phase (operating mode Warm\_Start or Cold\_Start)
  
- ❑ After this the regular function and process scheduling starts (operating mode Normal) but not immediately
  - ❑ To get a deterministic first release point of all processes the first activation is delayed until the first time window of the next period of the partition
    - This could be still in the same MAF or in the following one
    - Note: The standard claims only for the next period of the partition, but some OS ever delay the first release point in the next MAF
  
- ❑ After this point, the processes are activated according to their period (for periodic process)

# Partition Period Start – processes first release point

Window ID		2.1		2.2		2.3		2.4		2.5		2.1		2.2		2.3
Window Offset		20		70		95		120		170		20		70		95
Window Duration		30		15		15		30		30		30		15		15
Partition Period Start		t		t		f		t		t		t		t		f

Other\_TP

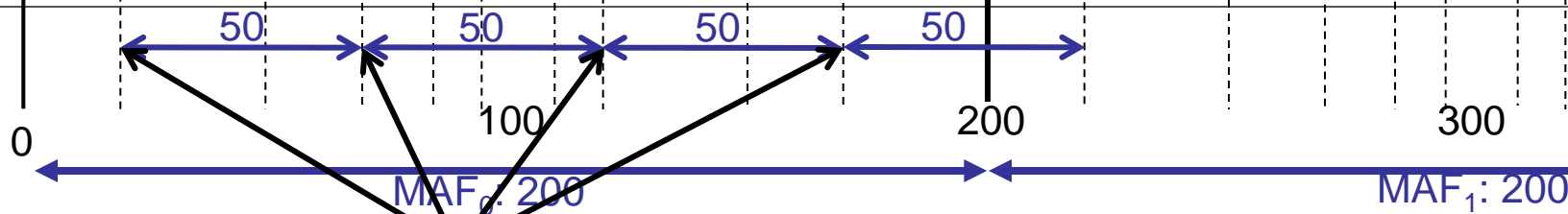
Prio 11

TP2\_P1 P50

D10 Prio10

TP2\_P2 P60

D15 Prio5



TP2: P: 50, D: 30

TP2: P:50, D:30  
Possible first release  
points in the MAF

MAF: Major Frame

TP: Time Partition

P: Period

D: Duration

Prio:Priority

# Partition Period Start

- If the processes periods are equal to the partition period, every next activations of all processes will occur in the same timing like the first one
  - If one or more process periods are not a multiple of the partition period, every next release point will be relative shifted than the earlier one
    - until it starts to repeat
  
- Because of the two reasons above the determination of the critical instance is hard and needs an automatic analysis

# Partition Period Start – processes first release point

Window ID		2.1		2.2		2.3		2.4		2.5		2.1		2.2		2.3
Window Offset		20		70		95		120		170		20		70		95
Window Duration		30		15		15		30		30		30		15		15
Partition Period Start		t		t		f		t		t		t		t		f

Other\_TP

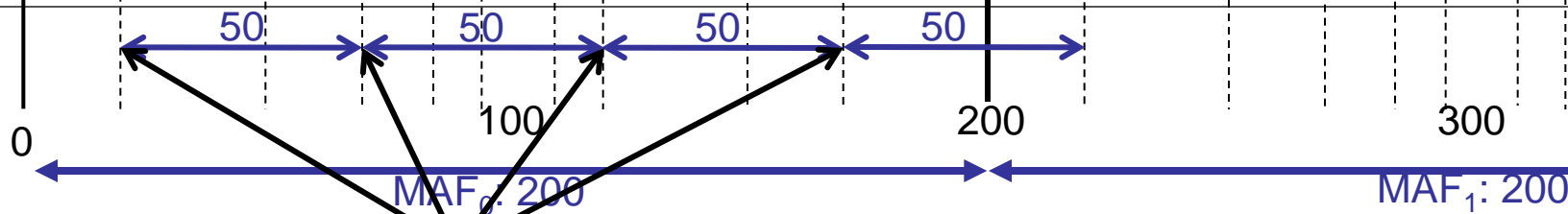
Prio 11

TP2\_P1 P50

D10 Prio10

TP2\_P2 P60

D15 Prio5



TP2: P: 50, D: 30

TP2: P:50, D:30  
Possible first release  
points in the MAF

MAF: Major Frame

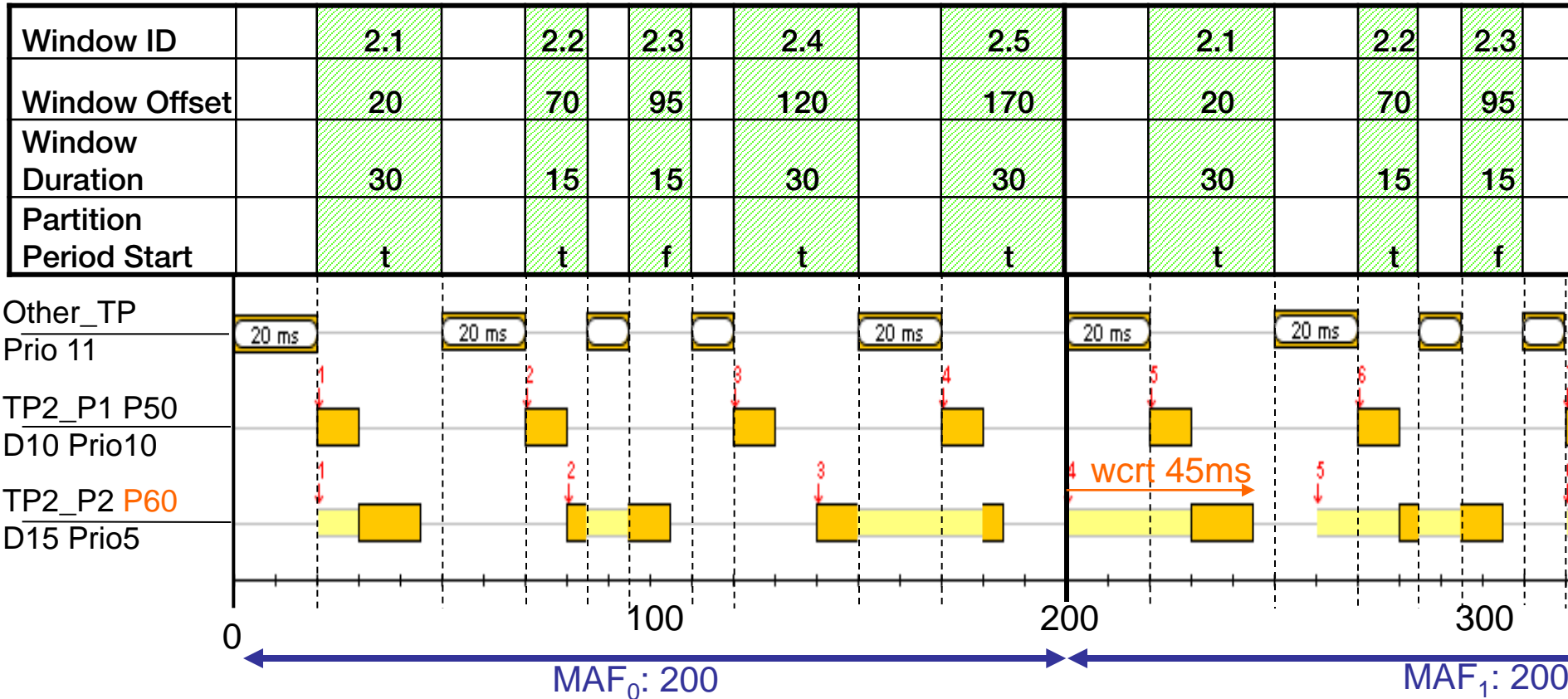
TP: Time Partition

P: Period

D: Duration

Prio: Priority

# Partition Period Start – processes first release point



TP2: P: 50, D: 30

MAF: Major Frame

TP: Time Partition

P: Period

D: Duration

Prio: Priority

# Partition Period Start – processes first release point

Window ID		2.1		2.2		2.3		2.4		2.5		2.1		2.2		2.3
Window Offset		20		70		95		120		170		20		70		95
Window Duration		30		15		15		30		30		30		15		15
Partition Period Start		t		t		f		t		t		t		t		f

Other\_TP

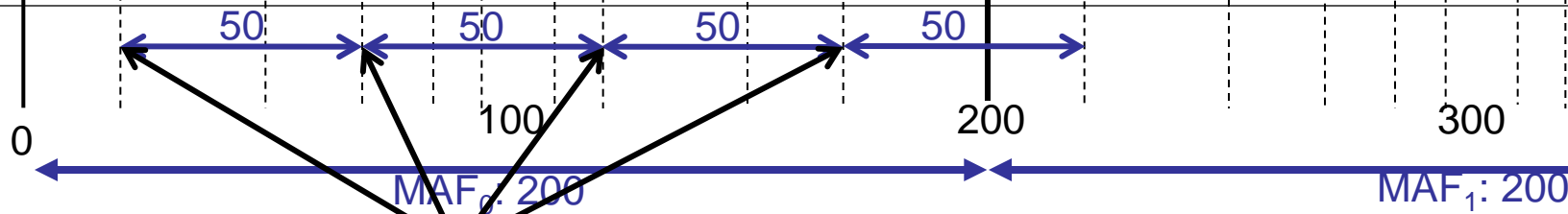
Prio 11

TP2\_P1 P50

D10 Prio10

TP2\_P2 P60

D15 Prio5



TP2: P: 50, D: 30

TP2: P:50, D:30  
Possible first release  
points in the MAF

MAF: Major Frame

TP: Time Partition

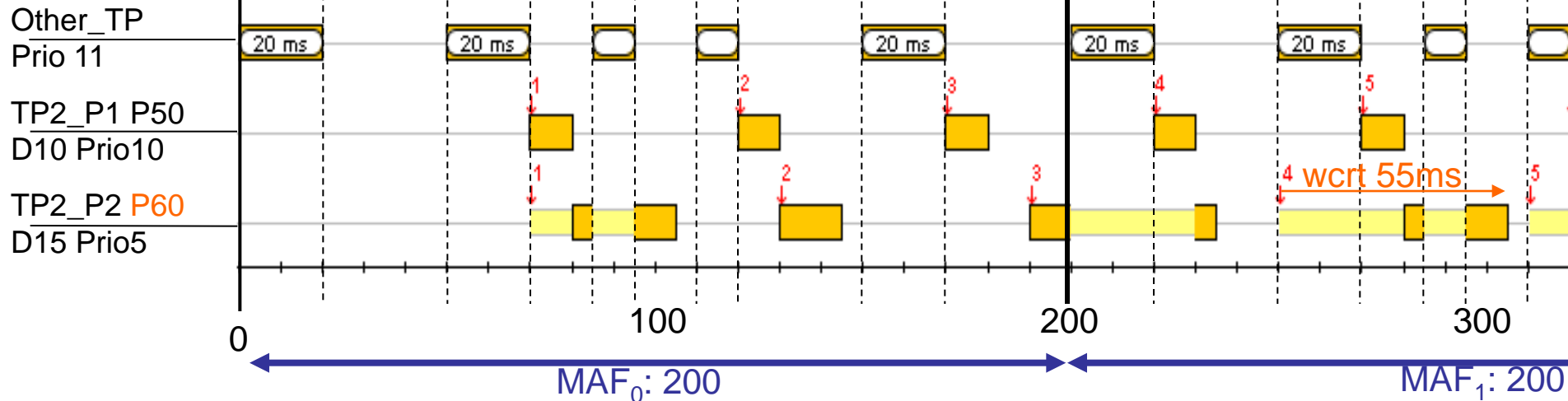
P: Period

D: Duration

Prio:Priority

# Partition Period Start – processes first release point

Window ID		2.1	2.2	2.3	2.4	2.5		2.1	2.2	2.3	
Window Offset		20	70	95	120	170		20	70	95	
Window Duration		30	15	15	30	30		30	15	15	
Partition Period Start		t	t	f	t	t		t	t	f	



TP2: P: 50, D: 30

MAF: Major Frame  
 TP: Time Partition  
 P: Period  
 D: Duration  
 Prio: Priority

# Dynamic Influences on ARINC 653 Scheduling



Solutions for Complex  
Real-Time Systems

# RTOS Overhead and Interrupts

## □ RTOS Overhead

- resource access, services

- communication drivers

  - (external interrupts from the network)

- error/exception handling (sporadic occurrence)

- context switch, between processes / partitions, (replenishment)

➔ Disturbs originally static schedule → “steals” WCET from applications !

□ Challenge: Guarantee capacity in presence of such disturbance!

➔ Requires scheduling analysis!

□ Symtavisision has expertise on static and dynamic systems, interrupts, etc. from automotive

# Design Flow

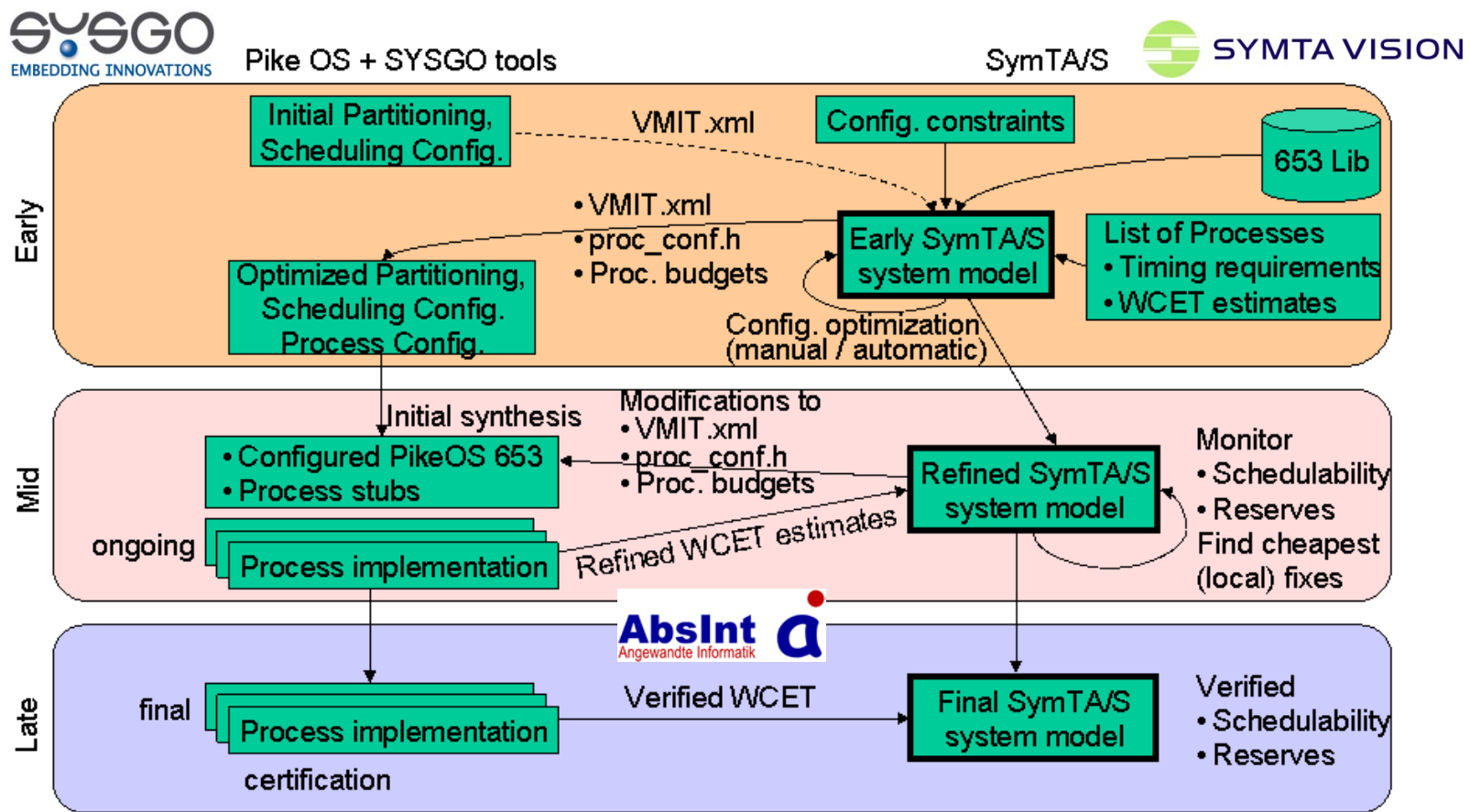


Solutions for Complex  
Real-Time Systems



# Prototype Design Flow (SYSGO & Symtavisision)

- Early: optimization of time-partitioning & initial intra-partition scheduling
- Mid: automatic configuration & code-generation based on optimized schedule
- Late: verify all real-time constraints; support certification & qualification



## Qualification as *Verification Tool* according to DO178B

- ❑ Customers can build confidence that SymTA/S provides right results for their operational environment with help of ***qualification support kit***
  - ❑ by executing a pre-defined set of analyses to exercise all meaningful input parameters
  - ❑ by measuring a representative set of problems and comparing to analysis results
  
- ❑ SymTA/S outputs reasoning, how it arrived at presenting a scheduling diagram as being a worst-case
  
- ❑ Supporting material: SymTA/S builds on mathematically *proven* foundation; Scientific papers are made available



SYMTA VISION

# Outlook on System-Level Timing

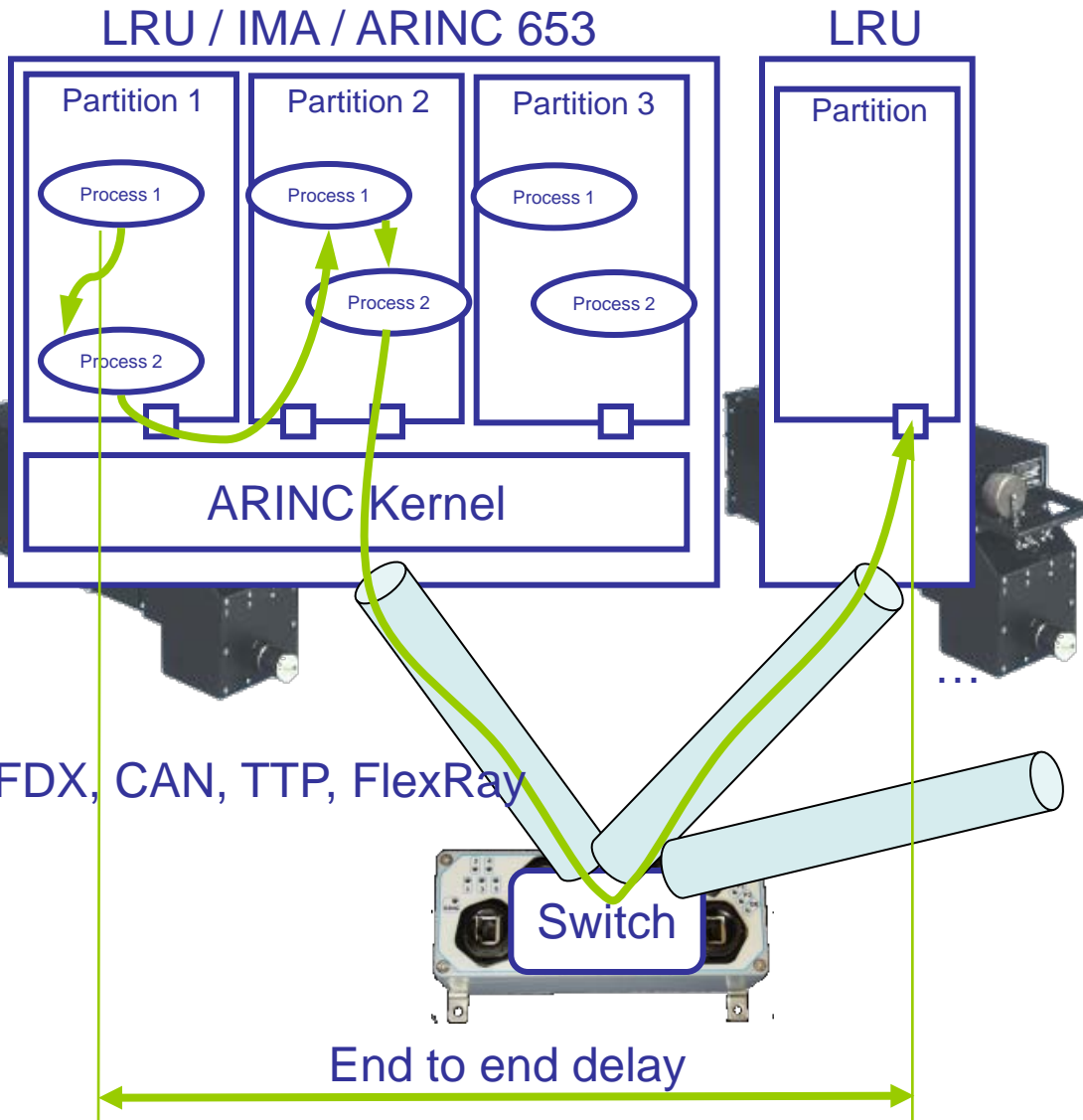
–

## ARINC 653 + TTP + AFDX Networks

Solutions for Complex  
Real-Time Systems



# System Timing Issues in Avionics Systems



Scope:

- ❑ single partition
- ❑ one LRU / IMA
- ❑ network end system
- ❑ switched network

Measure:

- ❑ load / utilization
- ❑ delays / deadlines
- ❑ communication / end-to-end

Use Case:

- ❑ design
- ❑ integration
- ❑ optimization
- ❑ verification

+ Design process integration

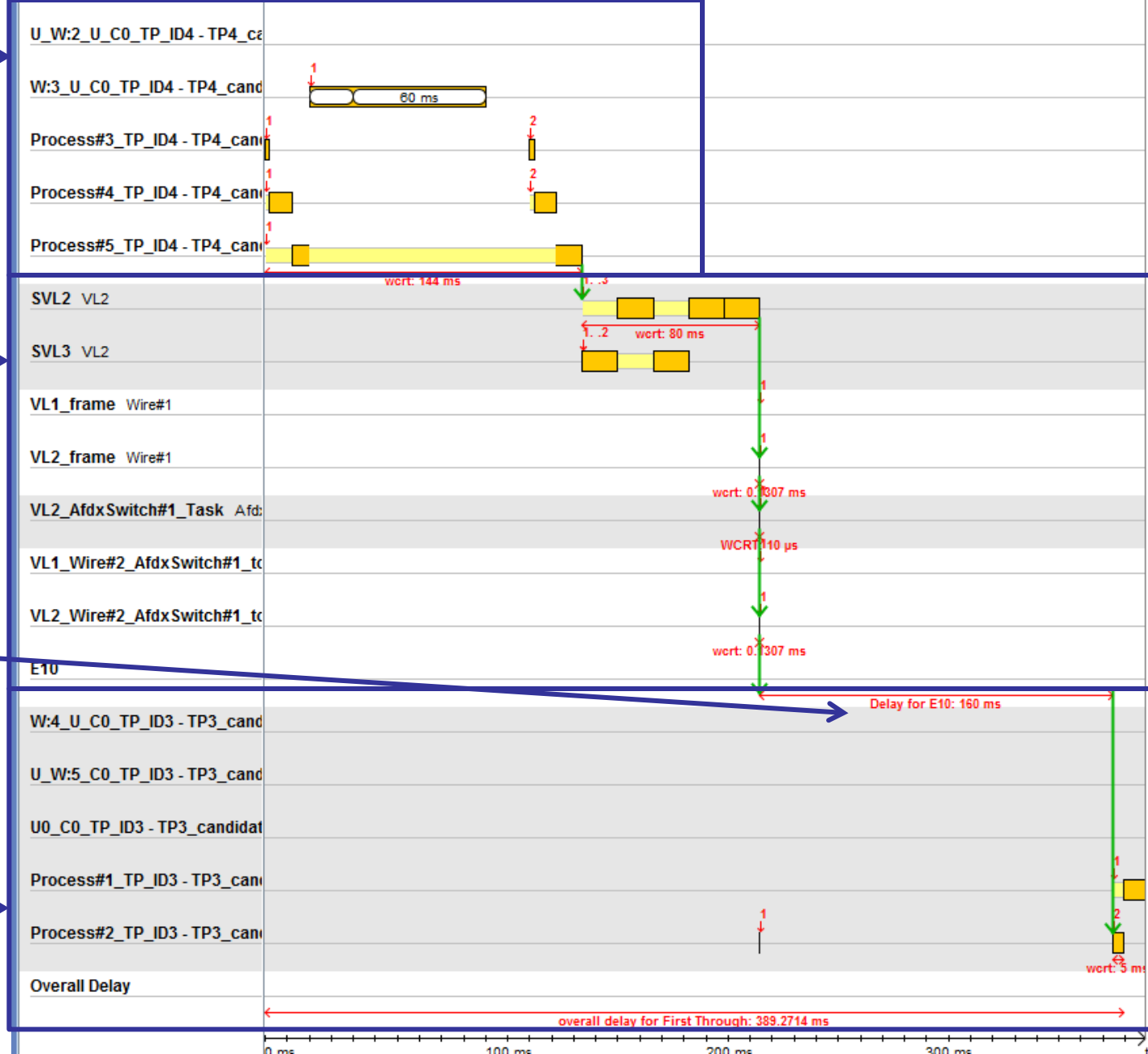
□ Sending IMA/Partition

□ AFDX

□ Sampling delay

→ Asynchronous execution

□ Receiving IMA/Partition



# Summary

Solutions for Complex  
Real-Time Systems



# Summary

- ❑ Virtual timing verification in early/mid phases avoid late integration problems
  - ❑ Refinement of timing requirements
  - ❑ First “virtual” verification steps
  - ❑ Cost reduction
  
- ❑ In late phases a detailed timing model for validation and verification is created
  - ❑ System level timing verification
  
- ❑ Tool support realized in SymTA/S
  
- ❑ Prototype integration with Sysgo exist

SymTA/S Project Explorer

- arinc
  - arinc\_11\_18\_05.xml
    - SymtaSystem
      - IMA
        - CPU
          - MajorFrame
            - TP 1 - KeySensor
              - OpenProcess
              - Qport1\_destination
              - Sport1\_destination
                - 1
                - 2
                - 3
                - 4
                - 50
            - TP 2 - DoorOpener
              - P1
              - Qport1\_source
              - Sport1\_source
                - 5
                - 6
                - 70
            - TP 3 - TP3
              - Process1
              - Process2
              - Sport3\_Destination
                - 7
                - 8
                - 90

MajorFrame

### Overview arinc\_11\_18\_05.xml MajorFrame MajorFrame

**MajorFrame**  
duration: 240.0 ms

**Parents**  
The element has the following parents  
CPU

**Element**  
name: MajorFrame

**Note**  
note:

**Possible Children Types**

Overview arinc\_11\_18\_05.xml MajorFrame MajorFrame

TP 3 - TP3

### TP Overview arinc\_11\_18\_05.xml TimePartition 3 - TP3

**TimePartition**  
identifier: 3  
Duration: 40.0 ms  
Period: 120.0 ms

**Parents**  
The element has the following parents  
CPU

**Element**  
name: 3 - TP3

**Possible Children Types**

Overview arinc\_11\_18\_05.xml TimePartition 3 - TP3

List of all WindowSchedules

	Element	WindowSchedule		
		name	windowOffset	windowDuration
1	1	0.0 ms	10.0 ms	1
2	2	25.0 ms	10.0 ms	2
3	3	40.0 ms	5.0 ms	3
4	4	65.0 ms	5.0 ms	4
5	5	10.0 ms	15.0 ms	5
6	6	45.0 ms	20.0 ms	6
7	7	70.0 ms	20.0 ms	7
8	8	100.0 ms	20.0 ms	8
9	50	120.0 ms	30.0 ms	50
10	70	150.0 ms	30.0 ms	70
11	90	190.0 ms	40.0 ms	90
12	new WindowSchedule	<empty>	<empty>	<empty>

WindowSchedules

Processes linked to TimePartition SymtaSystem.IMA.CPU.3 - TP3

	Element	Process				Resp...ime
		name	period	executionTime	timeCapacity	
1	Process1	120.0 ms	[15.0 ms;25.0 ms]	INF	4	<empty>
2	Process2	120.0 ms	[5.0 ms;10.0 ms]	INF	2	<empty>
3	new Process	<empty>	<empty>	<empty>	0	<empty>

Processes linked to TimePartition SymtaSystem.IMA.CPU.3 - TP3

Outline Tree View File As Root

type filter text

- Process1

List of all TimePartitions

	Element	TimePartition			Load
		name	identifier	Duration	
1	TP 1 - KeySensor	1	30.0 ms	120.0 ms	0.0
2	TP 2 - DoorOpener	2	30.0 ms	120.0 ms	0.0
3	TP 3 - TP3	3	40.0 ms	120.0 ms	0.0
4	new TimePartition	<empty>	<empty>	<empty>	0.0

TimePartitions

Process1

### Overview arinc\_11\_18\_05.xml Process Process1

**Process**  
period: 120.0 ms  
executionTime: [15.0 ms;25.0 ms]  
timeCapacity: INF  
priority: 4

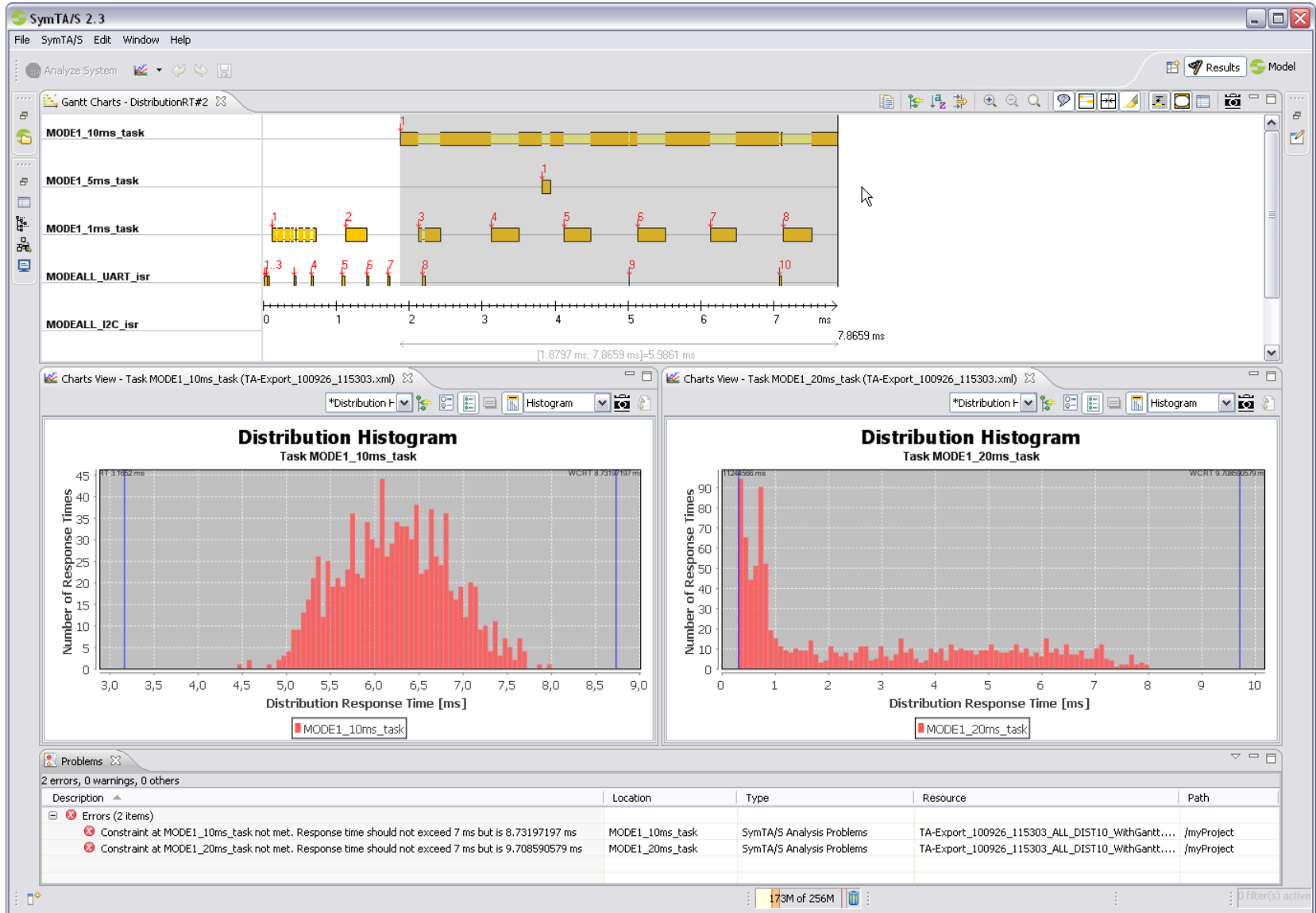
**Note**  
note:   
comment:   
logo:

**Parents**  
The element has the following parents  
TP 3 - TP3

**Element**  
name: Process1

Overview arinc\_11\_18\_05.xml Process Process1

# Best-Case, Worst-Case and Distribution Analysis



# Thank You!

[ficek@symtavisision.com](mailto:ficek@symtavisision.com)

[www.symtavisision.com](http://www.symtavisision.com)

Solutions for Complex  
Real-Time Systems

