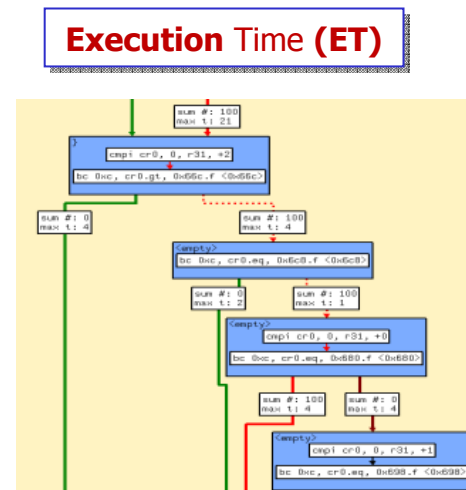
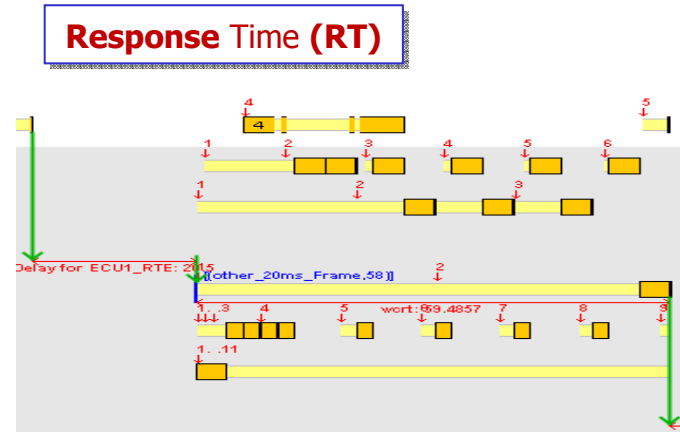


# An Integrated Timing Analysis Methodology for Real-Time Systems



# Dimension 1: System-Level and Code-Level

- System level
  - Many processes, tasks, communication (local, distributed)
  - Focus on
    - Integration and scheduling
    - Periodic or event-driven activation, blocking
    - End-to-end timing
  
- Code level
  - Single process, task, ISR
  - Focus on
    - Control flow
    - Processor architecture with pipelines and caches



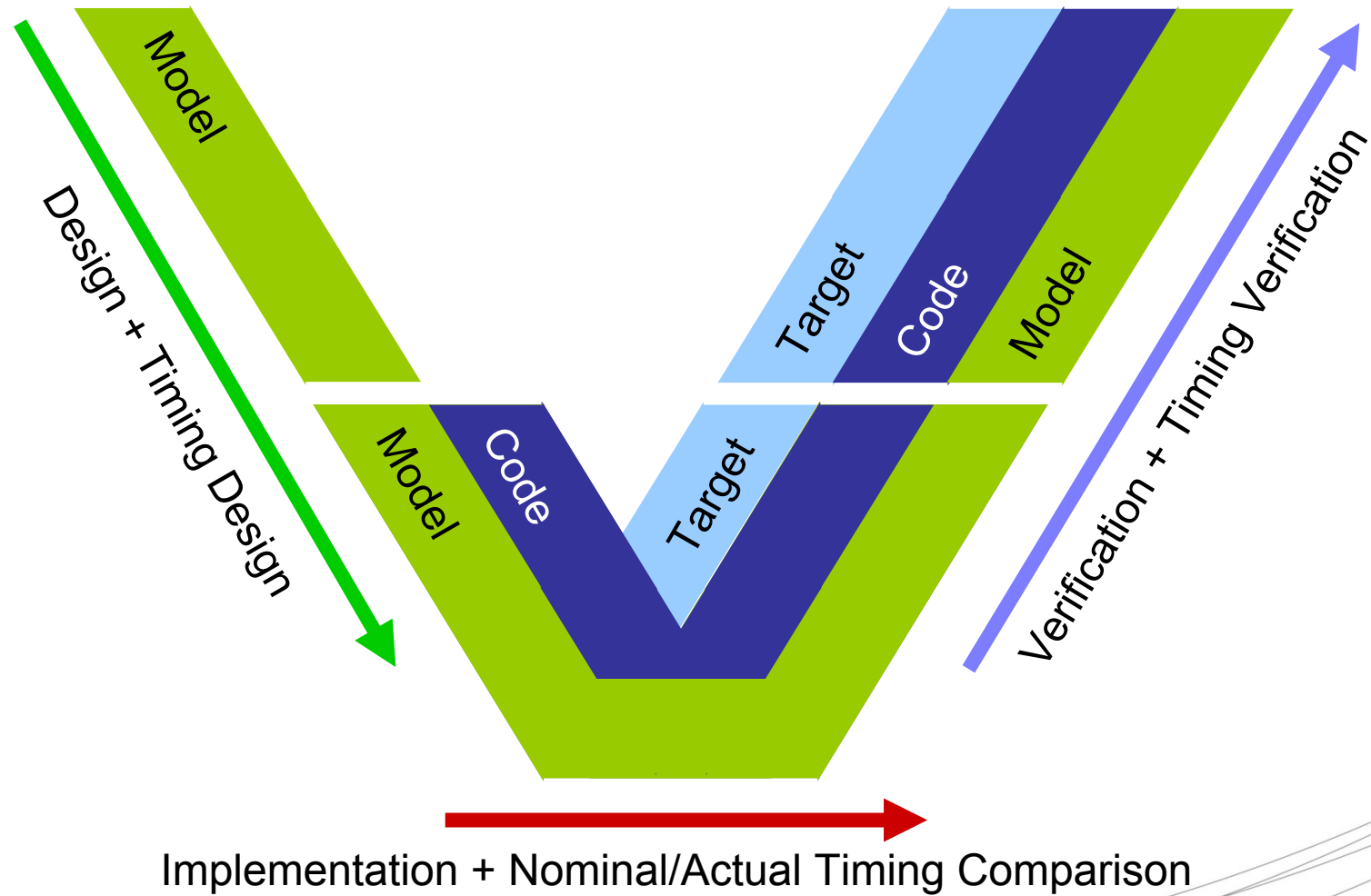


# Dimension 3: Criticality

- For safety-critical systems, **functional safety standards** require verification of **timing requirements**:
  - WCET: Worst-case execution time
  - WCRT: Worst-case response time
- **Static analysis** “recommended” / “highly recommended”
  - The static **aiT WCET Analyzer** has been used e.g. by **NASA** as an **industry-standard tool** for demonstrating the **absence of timing-related software defects** in the **Toyota** Unintended Acceleration Investigation.
  - The static **SymTA/S WCRT Analyzer** (Scheduling Analysis) is used e.g. as an **industry-standard tool** for verifying end-to-end deadlines in electromechanical steering ECUs.

Required by  
DO-178B / DO-178C /  
ISO-26262, EN-50128,  
IEC-61508

# Dimension 4: Design, Implementation and Verification



# WCET Analysis using Abstract Interpretation

Combines

- global static program analysis by **Abstract Interpretation**:  
microarchitecture analysis (caches, pipelines, ...) + value analysis
- integer linear programming for **path analysis**
- Example: aiT

Application Code

```
void Task (void) {
  variable++;
  function();
  next++;
  if (next)
    do this;
  terminate();
}
```

Compiler  
Linker

Executable (\*.elf / \*.out)

```
à ì ì @€
à ì @€
@ ì @€
kì @€
mì @€
ì @€
```

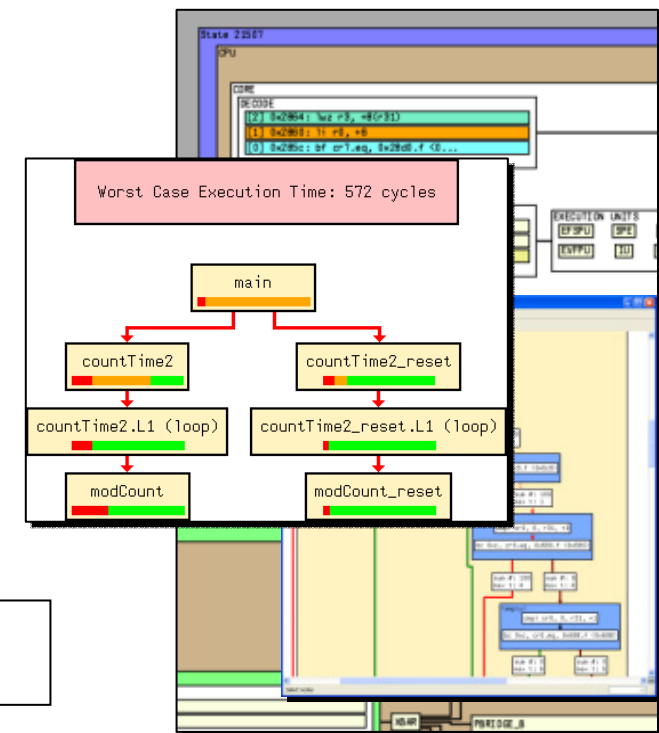
Specifications (\*.ais)

```
clock 10200 kHz ;
loop "_codebook" + 1 loop exactly 16 end ;
recursion "_fac" max 6;
SNIPPET "print" IS NOT ANALYZED AND TAKES MAX 333
CYCLES;
flow "U_MOD" + 0xAC bytes / "U_MOD" + 0xC4 bytes is max 4;
area from 0x20 to 0x497 is read-only;
```

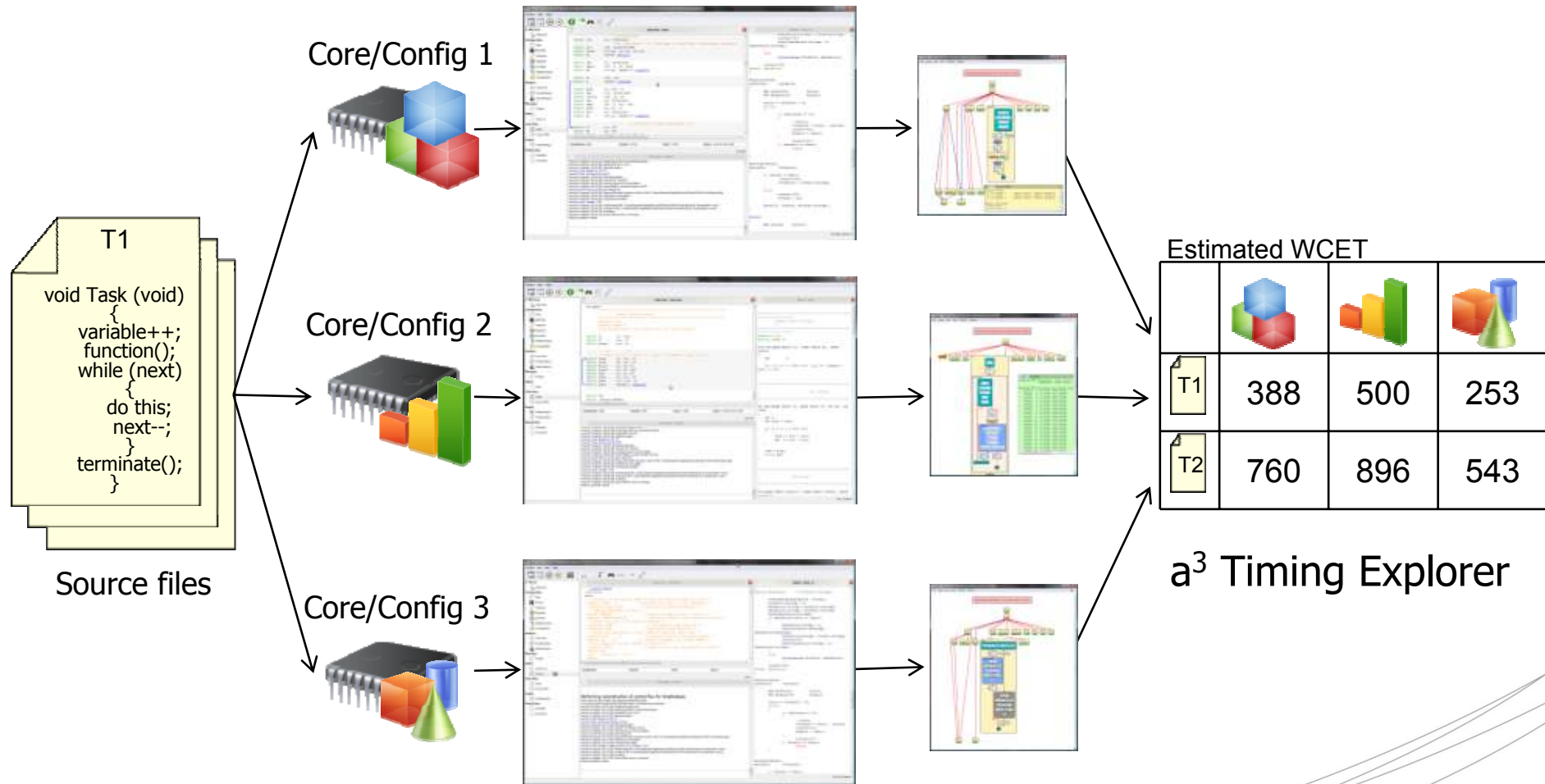
aiT

Entry Point

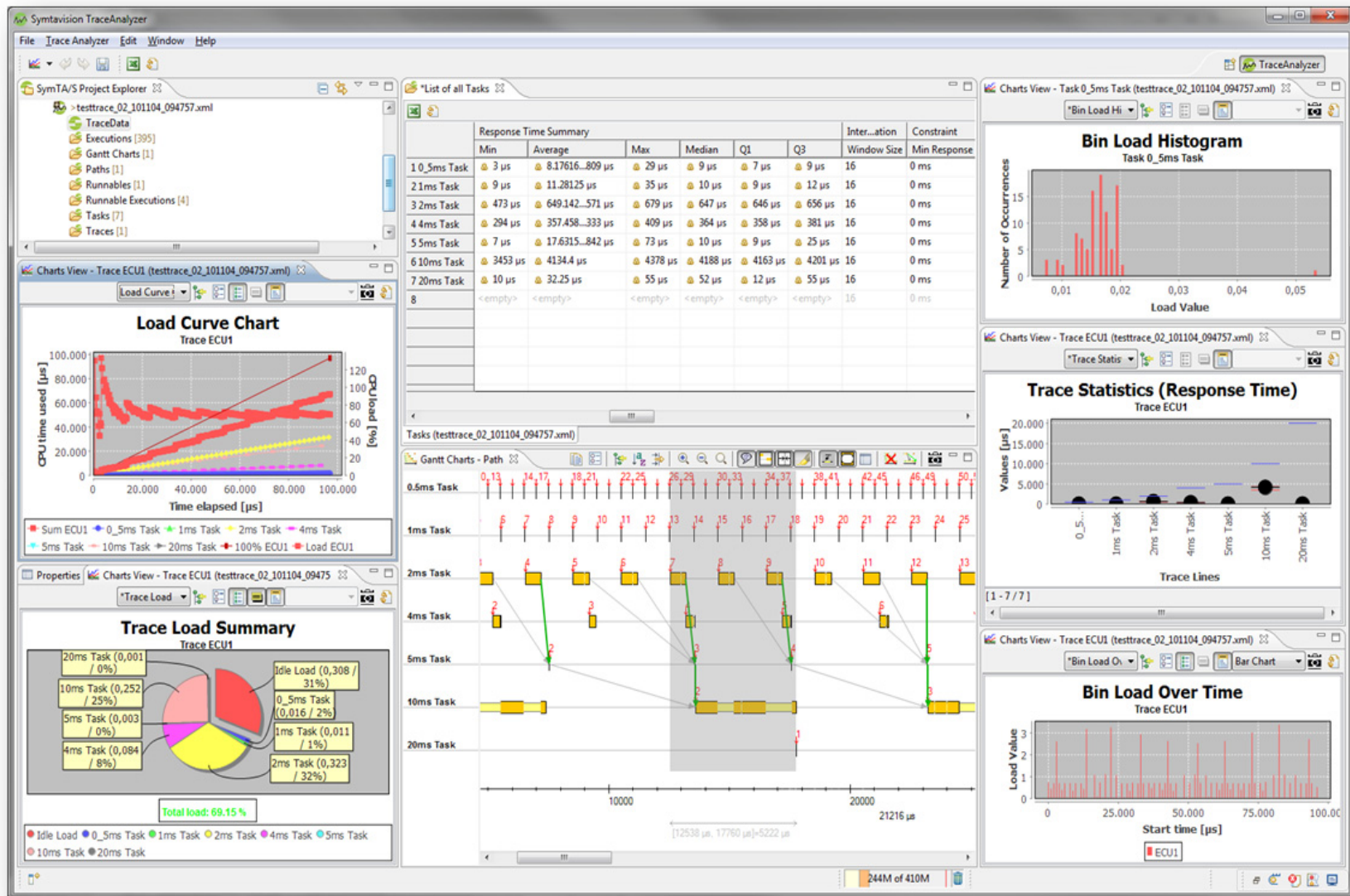
- Worst Case Execution Time
- Visualization, Documentation



# Code Estimation using Abstract Interpretation



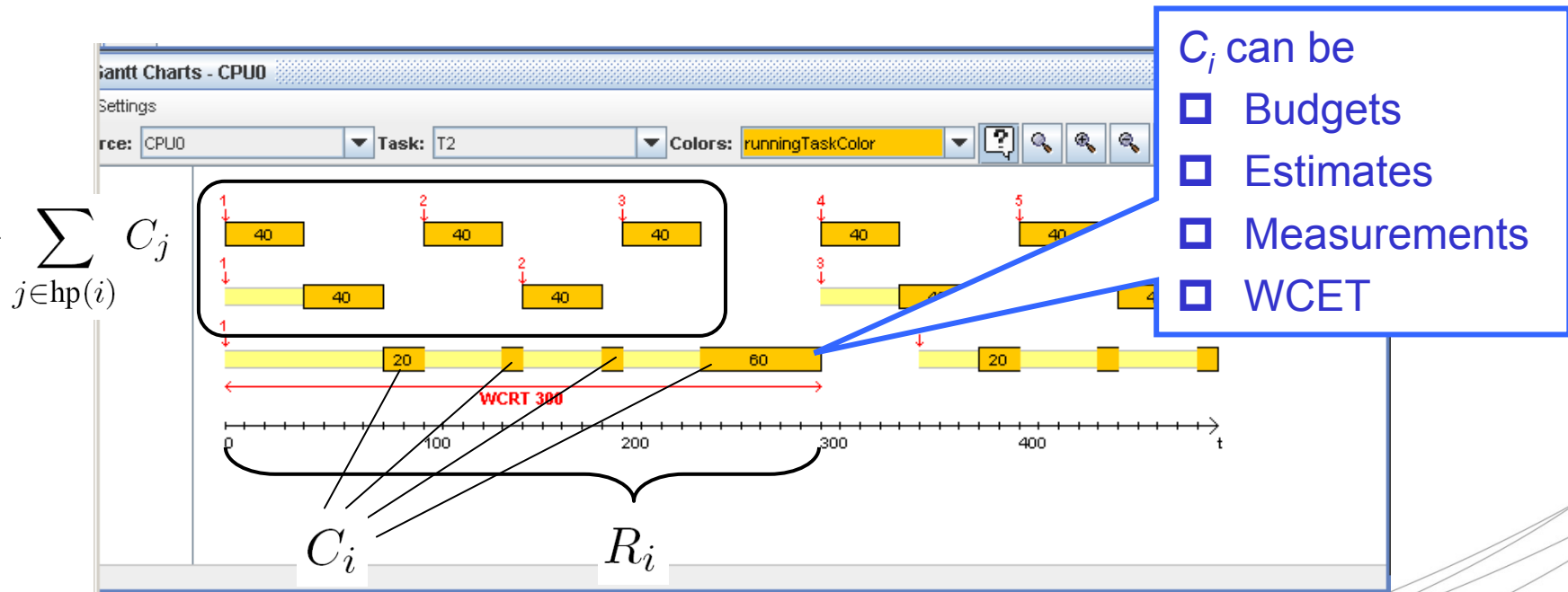
# Target Tracing



# Scheduling Analysis

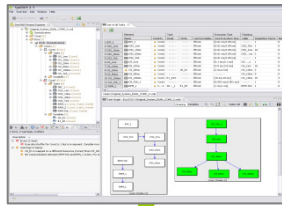
$$R_i = C_i + \underbrace{\sum_{j \in hp(i)} C_j \left\lfloor \frac{R_i}{T_j} \right\rfloor}_{\text{interference term } I_i}$$

Worst-Case Response Time (WCRT)      code execution time



# Scheduling Analysis, Distribution Analysis and Scheduling Simulation

*System Timing-Modell*



*User Input = bounds*

*User Input = fixed stimuli*

**Scheduling Simulation**

**Scheduling Analysis**

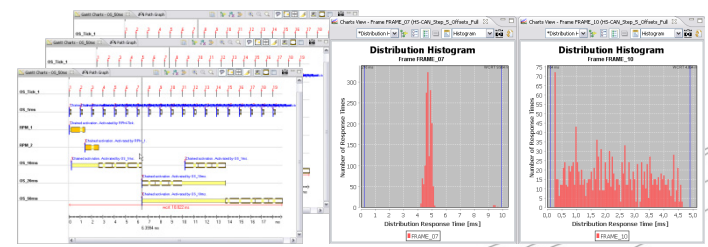
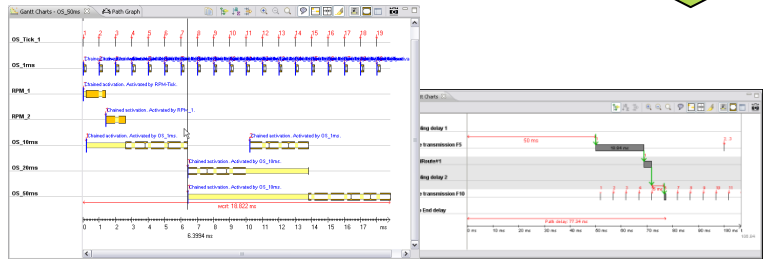
Worst-Case Stimuli

**Scheduling Simulation**

**Distribution Analysis**

Set of Stimuli

**Scheduling-Simulation**



SymTA/S

# Benefits

- Integrated methodology for the design, implementation and verification of embedded real-time systems
- Integrates system-level and code-level analysis tools
- Provides worst-case analysis and proof of functional safety through static WCET analysis and scheduling analysis
- Provides typical-case analysis through simulation, distribution analysis and tracing
- Enables seamless transition between model-based and code-based analysis
- Supports seamless flows from early system architecture exploration to final verification and certification

Thank you!

