The new ESA 32bit processor chipset is a fundamental building block for the next generation of space systems. Spacebel Informatique's Schedulability Analyser is a crucial tool that helps in developing and testing characterised and dependable software systems.

ERC32 Schedulability Analyser

Overview

The *Schedulability Analyser* provides a means to analytically verify, before execution time, that the timing constraints that apply to a real-time application are attainable. It also allows the user to assess the sensitivity of the system to changes in timing characteristics (task execution time, period, deadline) providing a means of tuning and increasing system efficiency and performance.

The Schedulability Analyser is used in an iterative way throughout the software life cycle of hard real-time applications. During the requirement and design phases the scheduling analysis is based on initial estimates of the proposed code execution time and on an initial model of thread interactions. These initial inputs are refined as the design process is carried out. Once the code is produced, a more accurate schedulability analysis can be performed.

As a result of the above activities, the design can either be reviewed or the Ada source code optimised. The highlight of areas with tight schedules and the analysis of missed deadlines are two indicators of the components to be redesigned or optimised. Sensitivity analysis allows the system designer to determine if optimisation is required and where to focus the efforts to obtain a schedulable system, therefore avoiding unnecessary or arbitrary optimisation.

The schedulability analysis is based on techniques determined from fixed priority scheduling theory. It uses an application model driven by the information required to apply these techniques.

The Schedulability Analyser also uses a computational model of the Ada runtime system to precisely quantify the system operations overheads.

The following fixed priority scheduling theories are supported by the tool:

- Fixed Priority Scheduling with deadline prior to period, commonly referred to as the Deadline Monotonic Scheduling theory.
- Fixed Priority Scheduling with Arbitrary deadlines or Arbitrary Deadline Scheduling.

Under both theories, threads only interact through Protected Objects. Protected objects ensure critical region access through a blocking protocol. Two blocking protocols are supported by the tool:

- Inhibit Interrupt,
- Immediate Priority Ceiling Inheritance.

Capability

The Schedulability Analyser report provides both system and detailed thread scheduling analysis.

At system level, the Schedulability Analyser provides the following information.

- Overall thread utilisation factor: the percentage of processor time spent in the execution of the overall thread set.
- Overall Schedulability: the schedulability of all the threads within the thread set.

At thread level, the Schedulability Analyser provides the following information.

- Schedulability: the result of the comparison of the thread deadline with the thread's worst case response time.
- Computed worst case computation time: the maximum processing time required by the thread for its execution. This time includes the run-time system overhead required to administrate a task.
- Worst case response time: the longest time taken by the thread to reply to a release. This time compared with the required deadline permits the schedulability of a task to be determined and the tightness of the schedule to be assessed.
- Maximum blocking time: the maximum time a thread is not provided access to the processor because of the execution of low priority threads with the identification of the conflicting shared resource.
- Margin sensitivity: the maximum increase in worst case computation time the thread may have and still remains schedulable or the minimum decrease in worst case computation time for the thread to become schedulable.

Interfaces

The Schedulability Analyser be used in one of two modes:

- GUI Mode where the user interacts with the tool through a GUI to select the data files, redirect warnings and errors, control the analysis. The analysis report is displayed in a separate window.
- Off-line Mode - the analysis is performed through a command line.

Example windows can be found on the backside of this data sheet.

The Schedulability Analyser requires the following information:

- the applicable scheduling theory and blocking protocol,
- a description of the application in terms of the application model described in the previous section,
- the runtime metrics quantifying the runtime system overhead required by the computational model.

This information is provided by three ASCII files:

- User Configuration File which defines the scheduling theory, blocking protocol, target hardware parameters, the thread real-time characteristics,
- Execution Skeleton File - the description of thread and protected object execution profiles which may be produced by the Thomson Software Products' Worst Case Computation Time Estimator tool,
- Run Time Characteristic File - the description of runtime metrics parts of the computational model(Manufactory supplied).

Models

ESA32 Schedulability Analyser

Producing dependable and predictable systems is a difficult task. The Schedulability Analyser provides a valuable help to assess the feasibility of the scheduling, starting at the specification phase.

Application Model

The schedulability analysis is based on an *Application Model* expressed in terms of scheduling entities and characteristics. The model defines the information required to perform the analysis:

- thread and protected object real-time characteristics,
- thread and protected object execution profiles.

The threads composing an application are of three sorts:

Cyclic - Cyclic threads execute at fixed time intervals (termed the period of the thread),

Sporadic - Sporadic threads execute under some random condition characterised with a finite non-zero minimum time between successive releases,

Interrupt Sporadic - Sporadic threads released by interrupt.

Threads are characterised by a

- *Deadline* - maximum allowable time from the thread release time until the required computation completion.
- *Criticality* - which may be
  - *Hard* - must meet their deadline otherwise catastrophic failure will result,
  - *Soft* - occasionally fail to meet their deadline without causing the system to fail,
  - *Non-Critical* - not subject to schedulability analysis.

  Thread criticality is the primary factor used for thread priority assignment. Hard threads are assigned highest priorities, followed by soft threads, then by non-critical threads. This ensures achievement of hard deadlines even in the case of soft thread overrun.
- *Period* for cyclic threads. *Minimum Inter-arrival time* for other threads.

Threads only interact through *Protected objects* and are either:

- *Resource control objects* - allows for data exchange between threads,
- *Synchronisation objects* - allows the release of a sporadic thread.

Protected Objects are accessed through a set of *Entry Points*. Each entry providing a critical region prohibiting other threads to enter the same region. A synchronisation object has two entry points.

- *Barriered Entry* - used to block the sporadic thread until its release. controlled by a *barrier*,
- *Non-Barriered Entry* - used to open the barrier and possibly release the waiting sporadic thread.

The model assumes that one synchronisation object only releases one sporadic thread.

Threads and protected objects are characterised by an *execution profile*. The execution profile of a thread or protected object is composed of a description of the *worst case execution path* of the thread or protected object entries and *the list of protected object entry calls* performed on others paths than the worst case execution path. The descriptions are used to compute accurate thread worst case execution and blocking times.

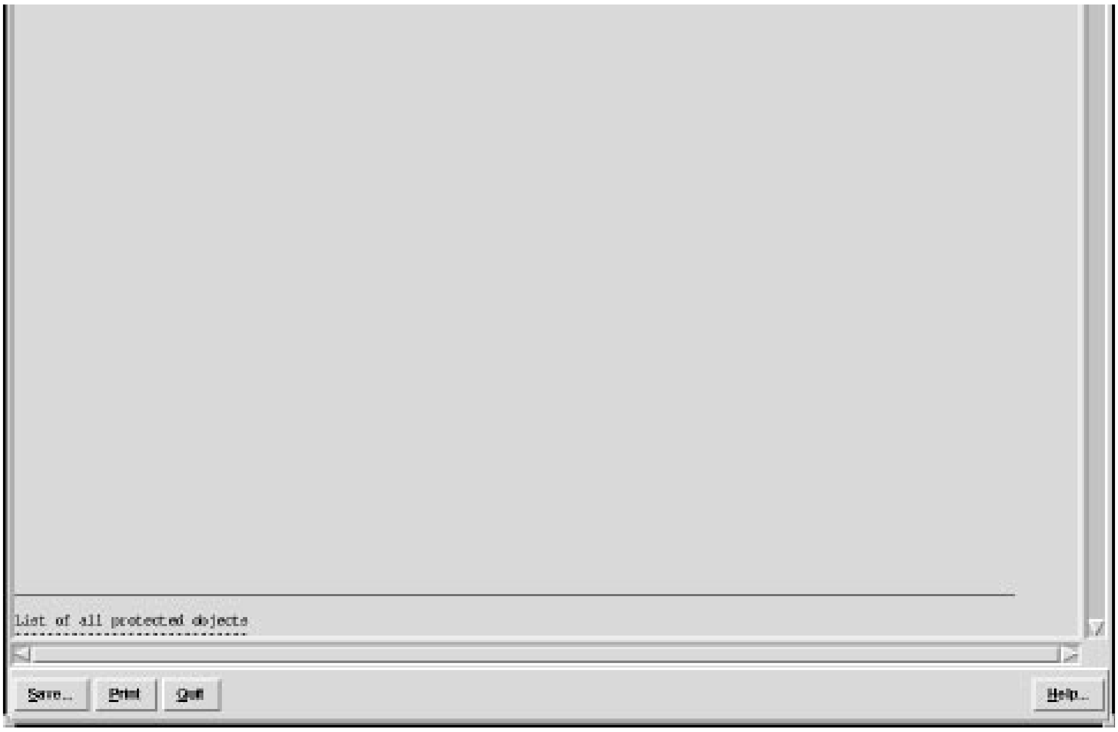
Computational Model

The computational model adopted for schedulability analysis takes a logical view of the implementation of the Ada runtime system. The model assumes

- Single processor system,
- Priority Pre-emptive Scheduler,
- Interval Timer driven clock,
- Delay queue, Ready Queue and Interrupt management

The model can be configure to handle two sorts of runtime systems with and without the ATAC.

The Ada runtime system is modelled by a number of metrics that quantify the cost of performing runtime system operations on behalf of the thread set. The metrics are used by the Schedulability Analyser to determine the interference of the Ada runtime system on the thread set.



Required Workstation

Any SUN workstation that runs Solaris 2.x and Open Windows or Motif, 8 MByte of free disk space. The display part of the product can be on another workstation or X-terminal.

It is believed that the contents of this data sheet is correct at publishing time. As the product is still evolving, future versions of the product might have slightly different characteristics and features.

Early access available through Estec evaluation programme.

More Information

web: http://www.spacebel.be (01/11/96)
email: esa32bit@spacebel.be
SPACEBEL Informatique s.a., 111, rue Colonel Bourg
1140 Brussels - Belgium,
Tel: +32 2 730 46 11 Fax: +32 2 726 85 13

A licensing scheme, maintenance and hot-line support is available.

SPACEBEL INFORMATIQUE s.a.

SPACEBEL Informatique is an engineering company that specialises in dependable, real-time, parallel, reusable object oriented systems for the various space segments