Incorporating The Notion of Importance into Mixed Criticality Systems

Tom Fleming
Department of Computer Science,
University of York, UK.
Email: tdf506@york.ac.uk

Alan Burns
Department of Computer Science,
University of York, UK.
Email: alan.burns@york.ac.uk

Structure

- Criticality change
- The impact on LO criticality tasks
- Importance
  - A simple Example
  - Analysis
- Importance VS Criticality
- Probability over an overrun
- Conclusion

The Criticality Change

Initial Mixed Criticality work did not expect tasks to exceed their LO budgets.

If we move to an assumption that criticality changes are likely to occur, we must consider two important issues.

- How can we reduce the impact a criticality change has on LO criticality tasks?
- How can we return to the LO criticality level?

This talk will consider a way to reduce the impact to LO criticality tasks.

The Criticality Change

Initial Mixed Criticality work did not expect tasks to exceed their LO budgets.

If we move to an assumption that criticality changes are likely to occur, we must consider two important issues.

- How can we reduce the impact a criticality change has on LO criticality tasks?
- How can we return to the LO criticality level?

This talk will consider a way to reduce the impact to LO criticality tasks.

The impact on LO Criticality tasks

If we assume that criticality changes are likely to occur.

- Suspending all LO criticality tasks is no longer acceptable.
- LO criticality tasks are still valuable.
- LO criticality tasks should only be suspended when absolutely necessary.
- The likelihood of a task executing to its full HI WCET is typically very low.

Can we exploit the slack in pessimistic WCET estimates for HI criticality tasks?

Importance

Importance defines an order in which tasks might be suspended providing a better level of service and graceful degradation for LO criticality tasks.

- Importance provides LO criticality graceful degradation
- Importance levels are assigned to tasks by the system designer.
- Tasks are suspended in order of importance.
- Points at which LO criticality tasks must be dropped are calculated offline.
- LO criticality tasks are allowed to run as for as long as possible.

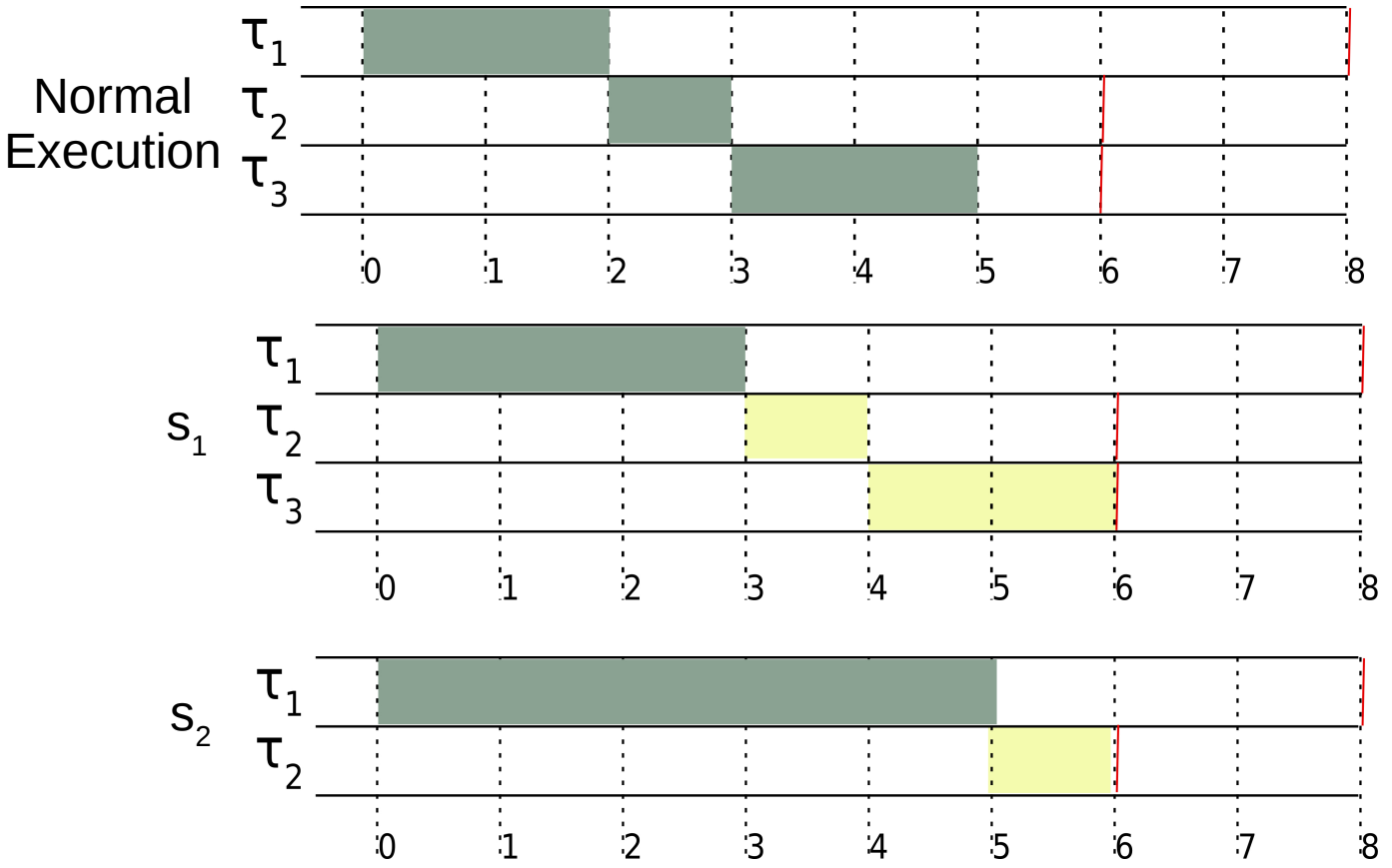
Simple Example

	C(LO)	C(HI)	T=D	L	P	I
τ_1	2	6	8	HI	1	-
τ_2	1	-	6	LO	2	1
τ_3	2	-	6	LO	3	2

We look to find a point of s indicating the severity of the HI criticality overrun at which the least important task currently executing must be suspended.

Although the LO criticality tasks are low priority, we provide a guarantee of service up to a certain level of overrun.

Simple Example



Importance Analysis

	C			L
τ_1	$C_1(LO)$	$C_1(ME)$	$C_1(HI)$	HI
τ_2	$C_2(LO)$	$C_2(ME)$	-	ME
τ_3	$C_3(LO)$	-	-	LO

The values of C(LO), C(ME) and C(HI) are given values.

These values represent the points, which if exceeded, a criticality change will occur.

This task set can be scheduled as a triple criticality system using the analysis from previous work (Extending mixed criticality scheduling, WMC 2013).

Importance Analysis

	C			L	I
τ_1	C_1(LO)	-	C_1(HI)	HI	-
τ_2	C_2(LO)	C_2(ME)	-	ME	1
τ_3	C_3(LO)	-	-	LO	2

Rather than use the provided value, we wish to calculate the point at which τ_2 must be suspended.

Importance Analysis

	C			L	I
τ_1	-	C(I1)	C_1(HI)	HI	-
τ_2	C_2(LO)	C_2(ME)	-	ME	1
τ_3	C_3(LO)	-	-	LO	2

The least important task, τ_3, may not need to be suspended at C_1(LO).

There may be slack which allow this task to be suspended later, we calculate this point.

Importance Analysis

	C			L	I
τ_1	C(I2)	C(I1)	C_1(HI)	HI	-
τ_2	C_2(LO)	C_2(LO)	-	LO	1
τ_3	C_3(LO)	-	-	LO	2

The resulting task set is no longer triple criticality.

It include the two calculated values C(I1) and C(I2) which represent the overrun required to suspended a level of importance.

A task, τ_4 , might be introduced that does not need to be suspended.

Assignment of Importance

Importance is not assigned in some optimal fashion, it is left to the system designer.

LO criticality, high importance tasks being dropped early in a HI criticality overrun could cause many LO tasks to be dropped in order to maintain the importance ordering.

A modified Audsley's approach attempts to give tasks with low importance low priorities.

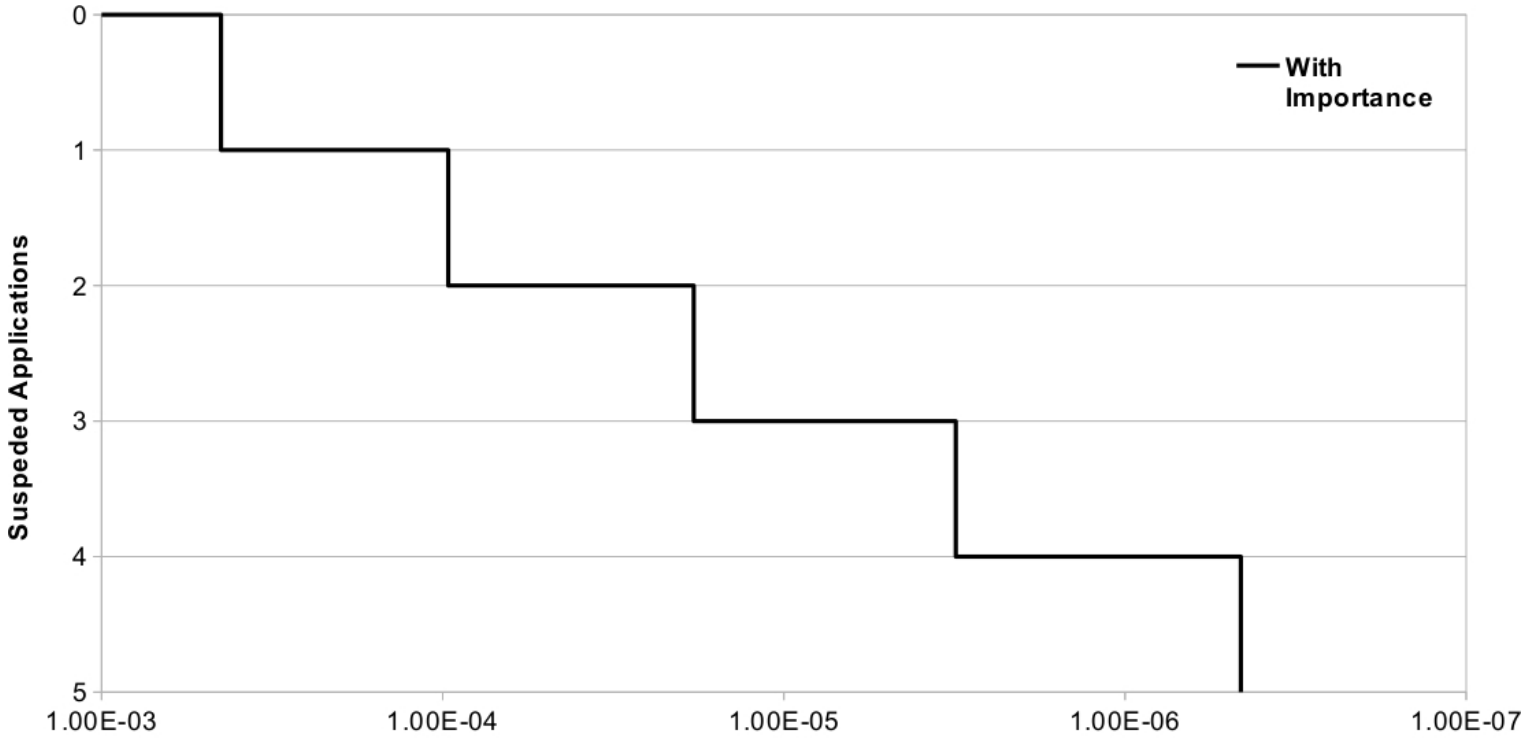
Importance and Criticality

Importance and Criticality are different assignments, although they do share common analysis.

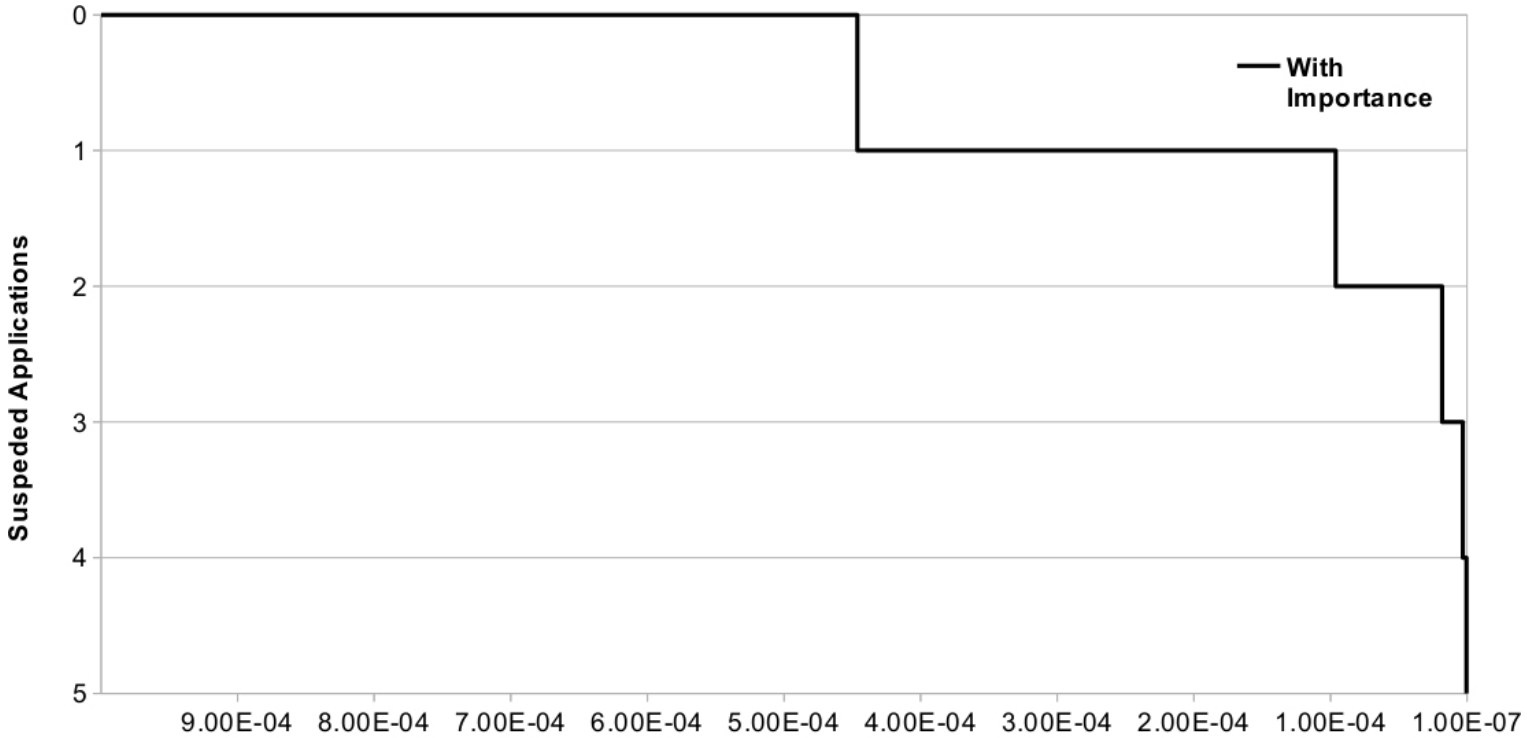
Criticality level: Often associated with a SIL (Safety Integrity Level) level, typically such a level is predefined by the type of software, e.g. Flight controller so Safety Critical.

Importance Level: Assigned by the system designer, provides the designer with some control over the degradation of the system.

Probability of an overrun



Probability of an overrun



Conclusions

Conclusions:

- Suspending all LO criticality tasks is too harsh.
- Introduced Importance to address this.
- Demonstrated how existing analysis can be used.
- Reasoned about the likelihood of a severe overrun.