Abstract

An incremental design approach for the execution architecture is described. The method is based on identification of the most critical requirement from both user as well as technical point of view. The implementation itself is based on quantified budgets. The creation, modification and verification of the budget is discussed.
Positioning in CAFCR

What does Customer need in Product and Why?

Customer What
- Customer objectives

Customer How
- Application

Product What
- Functional

Product How
- Conceptual
- Realization

SMART
- timing requirements
- external interfaces

execution architecture design
- threads
- allocation
- interrupts
- scheduling
- timers
- synchronization
- queues
- decoupling

models analysis

simulations measurements
Incremental approach

determine most important and critical requirements
model analyse constraints and design options
simulate build proto
measure evaluate analyse

An incremental execution architecture design approach
version: 1.0
August 21, 2020
EAAspiral
Decomposition of system TR in HW and SW

- **most and hardest TR handled by HW**
- **new control TRs**

**system TR**
- ns
- us
- ms
- s

**hardware TR**

**software TR**

An incremental execution architecture design approach

Gerrit Muller

version: 1.0
August 21, 2020
EAAhwswRequirements
Quantification steps

1. Order of magnitude
2. Guestimates
3. Calibrated estimates

- Back of the envelope
- Benchmark, spreadsheet calculation
- Measure, analyze, simulate
- Cycle accurate

Feasibility measure, analyze, simulate

Cycle accurate

99.999 → 100.001
Budget based design

An incremental execution architecture design approach

version: 1.0
August 21, 2020
EAAbudget

An incremental execution architecture design approach

micro benchmarks aggregated functions applications

measurements existing system

can be more complex than additions

model

design estimates; simulations

SRS

spec

t_{boot} 0.5s

t_{zep} 0.2s

measurements new (proto) system

feedback
tuning

budget

t_{proc} 10

t_{over} 20

T_{proc} 30

t_{disp} 5

t_{over} 20

T_{disp} 25

T_{total} 55

micro benchmarks aggregated functions applications profiles traces