Execution Architecture Soft Real Time design

by Gerrit Muller      Buskerud University College

e-mail: gaudisite@gmail.com

www.gaudisite.nl

Abstract

Distribution

This article or presentation is written as part of the Gaudi project. The Gaudi project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

March 6, 2013
status: planned
version: 0.2
Tension between different types of concerns

**Project management concerns**
- work-breakdown
- work allocation to competences or sites
- outsourcing
- purchasing
- anticipation on re-use
- maintainability

**Design choices**
- layering
- decoupling
- generic competence oriented decomposition structure

**Resource usage**
- memory
- CPU
- network
- disk

**Soft real time concerns**
- throughput
- latency
- response time

**Design choices**
- encryption
- authentication
- self sustained compression processing

**Quality concerns**
- security
- privacy
- robustness
- reliability
- AV quality

**Execution Architecture Soft Real Time design**
version: 0.2
March 6, 2013
EASRTension
Root causes of soft real time problems

code bloating ➔ resource bloating

3rd party SW, genericity

abundant layering or decomposition

too fine granularity   eg byte wise I/O

sequentialization

counterproductive optimization   eg prefetching

background activities   virus scanners, firewalls, polling activities   (Windows critical update)

scaleability of algorithm   e.g. searching brute force works upto ca 10000 entries
Performance as function of memory use

- performance
- memory usage
- physical memory
- paging to disk

Good

Bad

Memory usage

0 MB to 64 MB: Good
64 MB to 200 MB: Bad
Overhead penalty of modularity

- Modular
- Fine grain
- Value
- Overhead
- 81%
- Medium grain
- Monolithic
- Coarse grain
- Overhead
- 63%
- 44%
Function call overhead

Load and depth dependent (hidden) side effects
- pipeline flush
- I-cache disturbance
- D-cache disturbance

Prepare
- parameters
- save state
- jump

Access
- parameters

Do something useful

Return

Restore state

Do something useful

Legenda
- overhead
- value
Bloating explained

- Poor specification ("what")
- Poor design ("how")
- Dogmatic rules
  - For instance fine grain COM interfaces
- Genericity
- Configurability
- Provisions for future
- Support for unused legacy code

Legend:
- Overhead
- Value
Execution Architecture Soft Real Time design

Gerrit Muller

version: 0.2
March 6, 2013
EASRT_bloatingCausesBloating

Bloating causes more bloating

- Poor specification ("what")
- Poor design ("how")
- Decomposition overhead
- Support for unused legacy code

Legenda
- Overhead
- Value

Core functionality
- Configurability provisions for future
- Support for unused legacy code

Dogmatic rules
- Fine grain COM interfaces
- Dogmatic rules for instance fine grain COM interfaces

Genericity
- Poor specification ("what")
- Poor design ("how")
Bloating causes performance and resource problems. Solution: special measures: memory pools, shortcuts, ...

Blobbing causes even more bloating...

Execution Architecture Soft Real Time design
Gerrit Muller
March 6, 2013