Submethods in the CR Views

by Gerrit Muller  University of South-Eastern Norway-NISE

   e-mail: gaudisite@gmail.com
         www.gaudisite.nl

Abstract

This chapter describes the Conceptual view and the Realization view. Both views are supported by a set of submethods to describe multi-disciplinary design, for example several decompositions and models are provided.
Functional Decomposition

Submethods in the CR Views

version: 1.5
June 21, 2020
C VFuncti onal Decomposition
How about the **characteristic** of the **component** when performing **function**?

<table>
<thead>
<tr>
<th>characteristics</th>
<th>SNR</th>
<th>accuracy</th>
<th>memory usage</th>
<th>processing</th>
<th>latency</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>components</td>
<td>import server</td>
<td>user interface</td>
<td>print server</td>
<td>database server</td>
<td>export server</td>
<td>...</td>
</tr>
<tr>
<td>functions</td>
<td>query DB</td>
<td>render film</td>
<td>play movie</td>
<td>next</td>
<td>brightness</td>
<td>...</td>
</tr>
</tbody>
</table>

What is the **memory usage** of the **user interface** when **querying the DB**?
Process Decomposition

scan control
- scan UI
- scan control
- acq control
- recon control
- xDAS
- recon

image handling
- image handling UI
- db control
- archiving control
- import export
- display control
- display
- media
- network

legend
- UI process
- server process
- device hardware

Submethods in the CR Views
Gerrit Muller

version: 1.5
June 21, 2020
Conceptual Performance Model

\[ t_{\text{recon}} = t_{\text{filter}}(n_{\text{raw-x}}, n_{\text{raw-y}}) + n_{\text{raw-x}} \times (t_{\text{fft}}(n_{\text{raw-y}}) + n_{y} \times (t_{\text{fft}}(n_{\text{raw-x}}) + t_{\text{corrections}}(n_{x}, n_{y}) + t_{\text{control-overhead}}) + t_{\text{col-overhead}}) + t_{\text{row-overhead}}) + t_{\text{row-overhead}} + t_{\text{col-overhead}} + c_{\text{fft}} \times n \times \log(n) \]
Model After More Detailed Performance Analysis

Submethods in the CR Views

focus on overhead reduction

is more important

than faster algorithms

this is not an excuse for sloppy algorithms

version: 1.5
June 21, 2020
RVreconstructionPerformanceAnalysis
## Micro Benchmarks

<table>
<thead>
<tr>
<th>Submethods in the CR Views</th>
<th>Infrequent operations, often time-intensive</th>
<th>Often repeated operations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>database</strong></td>
<td>start session, finish session</td>
<td>perform transaction query</td>
</tr>
<tr>
<td><strong>network, I/O</strong></td>
<td>open connection, close connection</td>
<td>transfer data</td>
</tr>
<tr>
<td><strong>high level construction</strong></td>
<td>component creation, component destruction</td>
<td>method invocation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>same scope, other context</td>
</tr>
<tr>
<td><strong>low level construction</strong></td>
<td>object creation, object destruction</td>
<td>method invocation</td>
</tr>
<tr>
<td><strong>basic programming</strong></td>
<td>memory allocation, memory free</td>
<td>function call</td>
</tr>
<tr>
<td></td>
<td></td>
<td>loop overhead</td>
</tr>
<tr>
<td></td>
<td></td>
<td>basic operations (add, mul, load, store)</td>
</tr>
<tr>
<td><strong>OS</strong></td>
<td>task, thread creation</td>
<td>task switch, interrupt response</td>
</tr>
<tr>
<td><strong>HW</strong></td>
<td>power up, power down, boot</td>
<td>cache flush, low level data transfer</td>
</tr>
</tbody>
</table>
## Budget Approach

<table>
<thead>
<tr>
<th>step</th>
<th>example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1A measure old systems</strong></td>
<td>micro-benchmarks, aggregated functions, applications</td>
</tr>
<tr>
<td><strong>1B model the performance starting with old systems</strong></td>
<td>flow model and analytical model</td>
</tr>
<tr>
<td><strong>1C determine requirements for new system</strong></td>
<td>response time or throughput</td>
</tr>
<tr>
<td><strong>2 make a design for the new system</strong></td>
<td>explore design space, estimate and simulate</td>
</tr>
<tr>
<td><strong>3 make a budget for the new system:</strong></td>
<td>models provide the structure measurements and estimates provide initial numbers specification provides bottom line</td>
</tr>
<tr>
<td><strong>4 measure prototypes and new system</strong></td>
<td>micro-benchmarks, aggregated functions, applications profiles, traces</td>
</tr>
<tr>
<td><strong>5 Iterate steps 1B to 4</strong></td>
<td></td>
</tr>
<tr>
<td>Submethods in the CR Views</td>
<td>version: 1.5</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>(systematic) brainstorm</td>
<td>analysis and assessment</td>
</tr>
<tr>
<td>safety hazard analysis</td>
<td>potential hazards</td>
</tr>
<tr>
<td>reliability FMEA</td>
<td>failure modes</td>
</tr>
<tr>
<td>security</td>
<td>vulnerability risks</td>
</tr>
</tbody>
</table>
Work Break Down

TIP:NBE R1

scanning

viewing

database

computing

system

project organization

work packages

preparation conversion run time acq algorithms

xDAS xFEC reconstruction hardware

VDU console algorithms gfx UI

database engine clinical bulk data import export archive

host OS foundation classes start up shutdown exception handling

SPS SD TPS integration alfa test beta test conf man

legend

project

segment

make SW

make HW

buy SW

buy HW

system

version: 1.5
June 21, 2020
CVworkBreakdown
Integration Plan

2 partial systems for SW testing

2 existing base systems

new base systems

adopt existing base SW

new application

SW dev system

test and refine application

existing base system

integrate and refine application

SW for new HW subsystem

test SW for new HW subsystem

SW dev system

new HW subsystem

test HW subsystem

existing base system

integrate subsystem

adopt existing base SW

new base system

test new base system

integrate HW system

integrate system

application integration

new subsystem integration

integrated system

existing

new

version: 1.5
June 21, 2020
CVIntegrationPlan
## Overview CR Submethods

<table>
<thead>
<tr>
<th>Conceptual</th>
<th>Realization</th>
</tr>
</thead>
<tbody>
<tr>
<td>construction decomposition</td>
<td>budget</td>
</tr>
<tr>
<td>functional decomposition</td>
<td>benchmarking</td>
</tr>
<tr>
<td>designing with multiple</td>
<td>performance analysis</td>
</tr>
<tr>
<td>decompositions</td>
<td>value and cost</td>
</tr>
<tr>
<td>execution architecture</td>
<td>safety analysis</td>
</tr>
<tr>
<td>internal interfaces</td>
<td>reliability analysis</td>
</tr>
<tr>
<td>performance</td>
<td>security analysis</td>
</tr>
<tr>
<td>start up</td>
<td>granularity determination</td>
</tr>
<tr>
<td>shutdown</td>
<td></td>
</tr>
<tr>
<td>integration plan</td>
<td></td>
</tr>
<tr>
<td>work breakdown</td>
<td></td>
</tr>
<tr>
<td>safety</td>
<td></td>
</tr>
<tr>
<td>reliability</td>
<td></td>
</tr>
<tr>
<td>security</td>
<td></td>
</tr>
</tbody>
</table>

Submethods in the CR Views

13  Gerrit Muller

version: 1.5
June 21, 2020
TCRoverview