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.
Construction Decomposition

- **Applications**
  - view
  - PIP
  - adjust
  - view TXT

- **Services**
  - viewport
  - menu

- **Toolboxes**
  - audio
  - video
  - TXT
  - etc.

- **Driver**
  - drivers
  - scheduler
  - frame-buffer

- **Hardware**
  - tuner
  - MPEG
  - DSP

- **Domain Specific**
- **Generic**

- **Control Subsystem**
  - OS
  - CPU
  - RAM
  - etc.

- **File System**

- **Networking**

- **Browse**

Submethods in the CR Views

Version: 1.5
March 6, 2013
CVconstructionDecomposition
Functional Decomposition

Submethods in the CR Views

version: 1.5
March 6, 2013
CVfunctionalDecomposition

acquisition → acquisition processing → compress → encoding

encoding → storage

storage → decoding

display processing → display

decompress → de-compress → display processing

acquisition

processing

compress

encoding

storage

decoding
How about the <characteristic> of the <component> when performing <function>?
Submethods in the CR Views

Gerrit Muller

version: 1.5
March 6, 2013
CVprocessDecomposition
Conceptual Performance Model

\[
\begin{align*}
    t_{\text{recon}} &= t_{\text{filter}}(n_{\text{raw-x}}, n_{\text{raw-y}}) + n_{\text{raw-x}} \cdot (t_{\text{fft}}(n_{\text{raw-y}}) + n_{\text{y}} \cdot (t_{\text{fft}}(n_{\text{raw-x}}) + t_{\text{corrections}}(n_x, n_y) + t_{\text{control-overhead}}) + t_{\text{row-overhead}}) + t_{\text{col-overhead}})
\end{align*}
\]

\[
\begin{align*}
    t_{\text{fft}}(n) &= c_{\text{fft}} \cdot n \cdot \log(n)
\end{align*}
\]
Model After More Detailed Performance Analysis

\[
t_{\text{recon}} = t_{\text{filter}}(n_{\text{raw-x}}, n_{\text{raw-y}}) + n_{\text{raw-x}} \times (t_{\text{fft}}(n_{\text{raw-y}}) + t_{\text{col-overhead}}) + n_{y} \times (t_{\text{fft}}(n_{\text{raw-x}}) + t_{\text{row-overhead}}) + t_{\text{corrections}}(n_{x}, n_{y}) + t_{\text{read I/O}} + t_{\text{write I/O}} + t_{\text{control - overhead}}
\]

\[
t_{\text{fft}}(n) = c_{\text{fft}} \times n \times \log(n)
\]

Submethods in the CR Views

bookkeeping
transpose
malloc, free
write I/O
read I/O
overhead

correction computations
row overhead
FFT computations
column overhead
FFT computations
overhead
filter computations

focus on overhead reduction
is more important
than faster algorithms
this is not an excuse for sloppy algorithms
### Micro Benchmarks

<table>
<thead>
<tr>
<th>Category</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, 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, loop overhead, 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>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1A</td>
<td>Measure old systems</td>
<td>micro-benchmarks, aggregated functions, applications</td>
</tr>
<tr>
<td>1B</td>
<td>Model the performance starting with old systems</td>
<td>flow model and analytical model</td>
</tr>
<tr>
<td>1C</td>
<td>Determine requirements for new system</td>
<td>response time or throughput</td>
</tr>
<tr>
<td>2</td>
<td>Make a design for the new system</td>
<td>explore design space, estimate and simulate</td>
</tr>
<tr>
<td>3</td>
<td>Make a budget for the new system:</td>
<td>models provide the structure, measurements and estimates provide initial numbers, specification provides bottom line</td>
</tr>
<tr>
<td>4</td>
<td>Measure prototypes and new system</td>
<td>micro-benchmarks, aggregated functions, applications profiles, traces</td>
</tr>
<tr>
<td>5</td>
<td>Iterate steps 1B to 4</td>
<td></td>
</tr>
</tbody>
</table>
Safety, Reliability and Security Analysis

Submethods in the CR Views

10  Gerrit Muller

Safety, Reliability and Security Analysis

(systmatic) brainstorm → analysis and assessment → improve design

<table>
<thead>
<tr>
<th></th>
<th>potential hazards</th>
<th>probability</th>
<th>measures</th>
</tr>
</thead>
<tbody>
<tr>
<td>safety</td>
<td>hazard analysis</td>
<td>severity</td>
<td>measures</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>failure modes</th>
<th>effects</th>
<th>measures</th>
</tr>
</thead>
<tbody>
<tr>
<td>reliability</td>
<td>FMEA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>vulnerability risks</th>
<th>consequences</th>
<th>measures</th>
</tr>
</thead>
<tbody>
<tr>
<td>security</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Integration Plan

2 partial systems for SW testing

2 existing base systems

new base systems

1. adopt existing base SW
2. new application
3. SW dev system
4. test and refine application
5. existing base system
6. integrate and refine application
7. SW for new HW subsystem
8. test SW for new HW subsystem
9. SW dev system
10. new HW subsystem
11. test HW subsystem
12. integrate subsystem
13. existing base system
14. integrate and refine application
15. adopt existing base SW
16. new base system
17. test new base system
18. integrate HW system
19. integrate system

Submethods in the CR Views

version: 1.5
March 6, 2013
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 decompositions</td>
<td>performance analysis</td>
</tr>
<tr>
<td>execution architecture</td>
<td>value and cost</td>
</tr>
<tr>
<td>internal interfaces</td>
<td>safety analysis</td>
</tr>
<tr>
<td>performance</td>
<td>reliability analysis</td>
</tr>
<tr>
<td>start up</td>
<td>security analysis</td>
</tr>
<tr>
<td>shutdown</td>
<td>security analysis</td>
</tr>
<tr>
<td>integration plan</td>
<td>reliability analysis</td>
</tr>
<tr>
<td>work breakdown</td>
<td>security analysis</td>
</tr>
<tr>
<td>safety</td>
<td>granularity determination</td>
</tr>
<tr>
<td>reliability</td>
<td></td>
</tr>
<tr>
<td>security</td>
<td></td>
</tr>
</tbody>
</table>