Abstract

The concepts and realization of the medical imaging workstation are described. The following concepts are described: presentation and processing pipeline, resource management (CPU and memory), including caching and anti-fragmentation strategy, software process decomposition and decomposition rules.

The actual realization figures serve as illustration for the justification of some of the concepts.
Image Quality expectation WYSIWYG

X-ray system

image generation

presentation

monitor

film

network, storage

3rd party workstation

application processing

presentation

Easyvision

what you see at one work-spot is what you get at another work-spot

???
Presentation pipeline for X-ray images

- Image from database
- Spatial enhancement
- Interpolate (bi-linear, bi-cubic)
- Look up table (invert, contrast / brightness)
- Graphics merge
- Colour LUT
- Monitor

Legend:
- SW
- HW
Quadruple view-port screen layout

UI icons, text

view-port 1  view-port 2
view-port 3  view-port 4
view-port 5

960 pixels
1152 pixels
ca. 460 pixels
ca. 200 pixels

1152 pixels
1152 pixels
ca. 460 pixels
ca. 200 pixels
Rendered images at different destinations

**Screen:**
- low resolution
- fast response

**Film:**
- high resolution
- high throughput

**Network:**
- medium resolution
- high throughput
Concurrency via software processes

Remote systems and users

Communication

User

User interface

Data base

Export

Network

Disk drive

Optical storage

Optical disk drive

Print

Printer

Remote systems and users

User control

Client

Client process

System monitor

Unix daemons

Server process

Operational process

Associated hardware

Legend
Criterions for process decomposition

• management of concurrency
• management of shared devices
• unit of memory budget (easy measurement)
• enables distribution over multiple processors
• unit of exception handling: fault containment and watchdog monitor
Simplified layering of the software

Medical Imaging R/F

Print  Store  View  Cluster

Spool  HCU  Store  Image  Gfx  UI  DB  PMS-net in  PMS-net out

RC driver  HC driver  DOR driver  NIX

SunOS  Standard IPX workstation

Desk, cabinets, cables, etc.

Legend

user interface
application functions
toolbox
operating system
hardware
SW infrastructure
connected system

Dev. tools  service  SW keys  Config  Install  Start up

RC  3M  DSI
### Memory budget of Easyvision RF R1 and R2

<table>
<thead>
<tr>
<th>memory budget in Mbytes</th>
<th>code</th>
<th>object data</th>
<th>bulk data</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R1</td>
<td>R2</td>
<td>R1</td>
<td>R2</td>
</tr>
<tr>
<td>shared code</td>
<td>6.0</td>
<td>11.0</td>
<td>6.0</td>
<td>11.0</td>
</tr>
<tr>
<td>UI process</td>
<td>0.2</td>
<td>0.3</td>
<td>12.0</td>
<td>12.0</td>
</tr>
<tr>
<td>database server</td>
<td>0.2</td>
<td>0.3</td>
<td>3.0</td>
<td>3.0</td>
</tr>
<tr>
<td>print server</td>
<td>0.4</td>
<td>0.3</td>
<td>7.0</td>
<td>9.0</td>
</tr>
<tr>
<td>DOR server</td>
<td>0.4</td>
<td>0.3</td>
<td>2.0</td>
<td>1.0</td>
</tr>
<tr>
<td>communication server</td>
<td>1.2</td>
<td>0.3</td>
<td>10.0</td>
<td>4.0</td>
</tr>
<tr>
<td>UNIX commands</td>
<td>0.2</td>
<td>0.3</td>
<td>0.7</td>
<td>0.5</td>
</tr>
<tr>
<td>compute server</td>
<td>0.3</td>
<td>0.5</td>
<td>6.0</td>
<td>6.8</td>
</tr>
<tr>
<td>system monitor</td>
<td>0.3</td>
<td>0.5</td>
<td>0.8</td>
<td></td>
</tr>
<tr>
<td>application total</td>
<td>8.6</td>
<td>13.4</td>
<td>31.0</td>
<td>35.0</td>
</tr>
<tr>
<td>UNIX file cache</td>
<td></td>
<td></td>
<td>7.0</td>
<td>10.0</td>
</tr>
<tr>
<td>total</td>
<td></td>
<td></td>
<td>3.0</td>
<td>3.0</td>
</tr>
<tr>
<td>total</td>
<td></td>
<td></td>
<td>76.1</td>
<td>74.0</td>
</tr>
</tbody>
</table>
## Memory fragmentation

<table>
<thead>
<tr>
<th>image 1, 256 kB</th>
<th>image 2, 256 kB</th>
<th>image 3, 256 kB</th>
</tr>
</thead>
</table>

1. replace image 3 by image 4

<table>
<thead>
<tr>
<th>image 1, 256 kB</th>
<th>4</th>
<th>image 3, 256 kB</th>
</tr>
</thead>
</table>

2. add image 5

<table>
<thead>
<tr>
<th>image 1, 256 kB</th>
<th>4</th>
<th>image 3, 256 kB</th>
<th>image 5, 256 kB</th>
</tr>
</thead>
</table>

3. replace image 1 by image 6

<table>
<thead>
<tr>
<th>6</th>
<th>4</th>
<th>image 3, 256 kB</th>
<th>image 5, 256 kB</th>
</tr>
</thead>
</table>
Memory fragmentation increase

- used address space
- gross used
- nett used

MBytes

10
20

nett used
used address space
gross used

Medical Imaging Workstation: CR Views
11 Gerrit Muller
Cache layers

Medical imaging R/F cache sizes

- cluster PixMap cache
- print PixMap cache
- view PixMap cache
- allocator, chunk

heap memory, malloc() free()

virtual memory
- memory management unit
- instruction and data cache
- physical memory
- disk storage

Legend:
- user interface
- application functions
- toolbox
- operating system
- hardware
Bulk data memory management memory allocators

- **chunk size:** 3MB
  - for large images
    - from 225 kB (480*480*8)
    - to 3 MB (1536*1024*16)
  - **block size:** 256kB

- **chunk size:** 2MB
  - for small (screen) images
    - from 8kB
    - to 225 kB
  - **block size:** 8kB

- **chunk size:** 1MB
  - for stamp images
    - 96*96*8 (9kB)
  - **block size:** 9kB
Cached intermediate processing results

1. Retrieve raw image
2. Enhance image
3. Interpolate image
4. Look up grey-value image
5. Merge with view-port
6. Display the result
Example of allocator and cache use

Pixmap cache

- 1024^2 8 bit image requires 4 256kB blocks
- 8 1024^2 images require 48 256kB blocks
- 12 blocks shortage

- 4*1024^2 1 byte / pixel
- 4*1024^2 2 byte / pixel
- 4*460^2 2 byte / pixel
- 4*460^2 1 byte / pixel
- 4*460^2 1 byte / pixel

- block size: 256kB
- block size: 9kB
- block size: 8kB

460^2 image 8 bit requires 27 8kB blocks
200^2 images require 5 8kb blocks

all screen-size images require 334 8kB blocks, 78 blocks shortage

Medical Imaging Workstation: CR Views

version: 2.7
March 6, 2013
MICV pixmap example
Print server is based on banding

Medical Imaging Workstation: CR Views
version: 2.7
March 6, 2013
MICVbanding
CPU processing times and viewing responsiveness

Medical Imaging Workstation: CR Views

version: 2.7
March 6, 2013
MICVprocessingTimes

Embedded Systems Institute
Server CPU load

- Remote systems and users
- Communication
- Database
- Print
- Printer

- Serving one examination room:
  - 2.5 CPU second per Mbyte input
  - 3.5 CPU second per Mpixel output
  - CPU time available for interactive viewing
  - 30% of CPU time available

- Serving 3 examination rooms:
  - Import: 2.5 min per exam
  - Print: 10.5 min per exam
  - Margin: 2 min

- Remote systems and users:
  - 210 s per exam
  - 50 s per exam

Medical Imaging Workstation: CR Views
version: 2.7
March 6, 2013
Gerrit Muller
Resource measurement tools

\[ t_{n-2} \rightarrow \text{preamble to remove start-up effects} \rightarrow t_{n-1} \rightarrow \text{use case} \rightarrow t_n \rightarrow \text{time} \]

\[
\begin{align*}
oit & \quad \Delta \text{object instantations} \\
\text{ps} & \quad \text{heap memory usage} \\
\text{vmstat} & \quad \text{kernel CPU time} \\
\text{kernel resource stats} & \quad \text{user CPU time} \\
\end{align*}
\]

\[
\begin{align*}
\text{heapviewer (visualise fragmentation)}
\end{align*}
\]
## Object Instantiation Tracing

<table>
<thead>
<tr>
<th>class name</th>
<th>current nr of objects</th>
<th>deleted since $t_{n-1}$</th>
<th>created since $t_{n-1}$</th>
<th>heap memory usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>AsynchronousIO</td>
<td>0</td>
<td>-3</td>
<td>+3</td>
<td></td>
</tr>
<tr>
<td>AttributeEntry</td>
<td>237</td>
<td>-1</td>
<td>+5</td>
<td></td>
</tr>
<tr>
<td>BitMap</td>
<td>21</td>
<td>-4</td>
<td>+8</td>
<td></td>
</tr>
<tr>
<td>BoundedFloatingPoint</td>
<td>1034</td>
<td>-3</td>
<td>+22</td>
<td></td>
</tr>
<tr>
<td>BoundedInteger</td>
<td>684</td>
<td>-1</td>
<td>+9</td>
<td></td>
</tr>
<tr>
<td>BtreeNode1</td>
<td>200</td>
<td>-3</td>
<td>+3</td>
<td></td>
</tr>
<tr>
<td>BulkData</td>
<td>25</td>
<td>0</td>
<td>1</td>
<td>[819200]</td>
</tr>
<tr>
<td>ButtonGadget</td>
<td>34</td>
<td>0</td>
<td>2</td>
<td>[8388608]</td>
</tr>
<tr>
<td>ButtonStack</td>
<td>12</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>ByteArray</td>
<td>156</td>
<td>-4</td>
<td>+12</td>
<td>[13252]</td>
</tr>
</tbody>
</table>
### Overview of benchmarks and other measurement tools

<table>
<thead>
<tr>
<th>Test / Benchmark</th>
<th>What, Why</th>
<th>Accuracy</th>
<th>When</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Public</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SpecInt (by suppliers)</td>
<td>CPU integer</td>
<td>coarse</td>
<td>new hardware</td>
</tr>
<tr>
<td>Byte benchmark</td>
<td>computer platform performance OS, shell, file I/O</td>
<td>coarse</td>
<td>new hardware new OS release</td>
</tr>
<tr>
<td><strong>Self made</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>file I/O</td>
<td>file I/O throughput</td>
<td>medium</td>
<td>new hardware</td>
</tr>
<tr>
<td>image processing</td>
<td>CPU, cache, memory as function of image, pixel size</td>
<td>accurate</td>
<td>new hardware</td>
</tr>
<tr>
<td>Objective-C overhead</td>
<td>method call overhead memory overhead</td>
<td>accurate</td>
<td>initial</td>
</tr>
<tr>
<td>socket, network</td>
<td>throughput CPU overhead</td>
<td>accurate</td>
<td>ad hoc</td>
</tr>
<tr>
<td>data base</td>
<td>transaction overhead query behaviour</td>
<td>accurate</td>
<td>ad hoc</td>
</tr>
<tr>
<td>load test</td>
<td>throughput, CPU, memory</td>
<td>accurate</td>
<td>regression</td>
</tr>
</tbody>
</table>
Coverage of submethods of the CR views

<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>granularity determination</td>
</tr>
<tr>
<td>internal interfaces</td>
<td>value and cost</td>
</tr>
<tr>
<td>performance</td>
<td>safety analysis</td>
</tr>
<tr>
<td>start up</td>
<td>reliability analysis</td>
</tr>
<tr>
<td>shutdown</td>
<td>security analysis</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>

**legend**
- *explicitly addressed*
- *addressed only implicitly*
- *not addressed*

Coverage based on documentation status of first product release.
The case material is based on actual data, from a complex context with large commercial interests. The material is **simplified** to increase the accessibility, while at the same time **small changes** have been made to remove commercial sensitivity. Commercial sensitivity is further reduced by using relatively **old** data (between 5 and 10 years in the past). Care has been taken that the illustrative value is maintained.