Why Quantified Insight in System Design is Required.

by Gerrit Muller   Buskerud University College
   e-mail: gaudisite@gmail.com
   www.gaudisite.nl

Abstract
Software education is mostly function-oriented. Techniques and formalisms are focused on system behavior. Software architects often don’t have a quantified insight in problem domain or chosen solutions, although computers work internally with bits and bytes. This is a problem for IT systems in general, but is more so for embedded systems. Embedded systems interact with the physical world, which can be modeled quantitatively: energy consumption, speed, force, et cetera. This presentation addresses quantification of system and software design, illustrated by case examples.
Why Quantified Insight in System Design is Required.

Gerrit Muller

version: 0.2
March 6, 2013
QSDpurposes
Challenge

Ask a SW-architect to quantify the product under construction.

What happens?
Ask a SW-architect to *quantify* the product under construction.

What happens?

The *project* is quantified, rather than the *system* of interest

| man-years | code-complexity |
| lines-of-code | fault density |
| problem reports | release schedule |
The SW engineering discipline today is **process** oriented, quantities are process metrics.

The System Of Interest (SOI) is designed from **behavioral** point of view.

Conventional Engineering disciplines design the SOI with **quantitative** techniques.

Qualities of SW intensive systems, such as performance, are **emerging** i.s.o. **predictable** properties.
example of conventional engineering: control engineering

performance engineering: example of quantified SW design

where and how do control and SW engineering meet?

reflection
Why Quantified Insight in System Design is Required.
Measuring Disturbance Transfer

A\textsubscript{out} \over A\textsubscript{in}

controller + motor

A\textsubscript{in} disturbance

A\textsubscript{out} disturbance

Why Quantified Insight in System Design is Required.

Gerrit Muller

version: 0.2
March 6, 2013
PENGcontrolDisturbanceMeasurement
Idealized Disturbance Transfer

Why Quantified Insight in System Design is Required.

version: 0.2
March 6, 2013
PENGcontrolDisturbanceTransfer
Measuring Tracking Response

Why Quantified Insight in System Design is Required.

Gerrit Muller

version: 0.2
March 6, 2013
PENGcontrolTrackMeasurement
Idealized Tracking Response

Why Quantified Insight in System Design is Required.

Gerrit Muller

version: 0.2
March 6, 2013
PENgcontrolTrackTransfer
Black Box Model

stimulus $\rightarrow$ black box $\rightarrow$ response

black box:
- simplified model
- mathematical formula
- with physical interpretation:
  response = f(stimulus, parameters)

parameters:
- bandwidth
- overshoot
- order
- ...

Why Quantified Insight in System Design is Required.

version: 0.2
March 6, 2013
PENGblackBox
White Box Model

stimulus → white box
   friction
   hysteresis
   interference
   ...

parameters
   bandwidth
   overshoot
   order
   ...

↑

response

white box :
black box model plus
experience based reasoning
over non-idealities
response = f(stimulus, parameters,
some parametrized non-idealities)

challenge: to know what non-idealities to ignore
and to ignore as much as possible
Control Engineering Knowledge

Why Quantified Insight in System Design is Required.

version: 0.2
March 6, 2013
PENGcontrolKnowledge
Summary of Control Engineering

experience → consolidation and reflection → formalisms

research → consolidation and reflection → techniques

consolidation and reflection → methods

models → robustness

representations → design for ....

measurements
example of conventional engineering: control engineering
performance engineering: example of quantified SW design
where and how do control and SW engineering meet?
reflection
What is the Performance of this Code?

Sample application code:

```
for x = 1 to 3 {
    for y = 1 to 3 {
        retrieve_image(x,y)
    }
}
```

alternative application code:

```
event 3*3 -> show screen 3*3

<screen 3*3>
    <row 1>
        <col 1><image 1,1></col 1>
        <col 2><image 1,2></col 2>
        <col 3><image 1,3></col 3>
    </row 1>
    <row 2>
        <col 1><image 1,1></col 1>
        <col 2><image 1,2></col 2>
        <col 3><image 1,3></col 3>
    </row 2>
    <row 3>
        <col 1><image 1,1></col 1>
        <col 2><image 1,2></col 2>
        <col 3><image 1,3></col 3>
    </row 3>
</screen 3*3>
```

application need:

at event 3*3 show 3*3 images instanteneous
What if . . .

What If....

Sample application code:

```plaintext
for x = 1 to 3 {
    for y = 1 to 3 {
        retrieve_image(x,y)
    }
}
```

Why Quantified Insight in System Design is Required.

Gerrit Muller
What if . . .

What If....

Sample application code:

```plaintext
for x = 1 to 3 {
    for y = 1 to 3 {
        retrieve_image(x, y)
    }
}
```

Why Quantified Insight in System Design is Required.
What If....

Sample application code:

```java
for x = 1 to 3 {
    for y = 1 to 3 {
        retrieve_image(x,y)
    }
}
```

Attribute = 1 COM object
100 attributes / image
9 images = 900 COM objects
1 COM object = 80µs
9 images = 72 ms
What If....

- I/O on line basis ($512^2$ image)

  \[ 9 \times 512 \times t_{I/O} \]

  \[ t_{I/O} \sim= 1 \text{ms} \]
Challenge SW Performance Design

<table>
<thead>
<tr>
<th>F &amp; S</th>
<th>F &amp; S</th>
<th>F &amp; S</th>
<th>F &amp; S</th>
<th>F &amp; S</th>
<th>F &amp; S</th>
<th>F &amp; S</th>
</tr>
</thead>
<tbody>
<tr>
<td>MW</td>
<td>MW</td>
<td>MW</td>
<td>MW</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OS</td>
<td>OS</td>
<td>OS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HW</td>
<td>HW</td>
<td>HW</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Functions & Services
Middleware
Operating systems
Hardware

Performance = Function (F&S, other F&S, MW, OS, HW)
MW, OS, HW >> 100 Manyear: very complex

Challenge: How to understand MW, OS, HW with only a few parameters
Layered Benchmarking

**typical values**
*interference*
*variation*
*boundaries*

---

**CPU**
**cache**
**memory**
**bus**
.. (computing) hardware

typical values
interference
variation
boundaries

---

**duration**
**footprint**
**interrupts**
**task switches**
**OS services**
**CPU time**
**footprint**
**cache**

---

**network transfer**
**database access**
**database query**
services/functions

---

**end-to-end function**

---

**applications**

---

**services**

---

**operating system**

---

**tools**

---

Why Quantified Insight in System Design is Required.

---

Gerrit Muller

---

version: 0.2
March 6, 2013
EBMlbenchmarkStack
Why Quantified Insight in System Design is Required.

Gerrit Muller
Example Hardware Performance

Memory access time in case of a cache miss
200 Mhz, 5 ns cycle: 190 ns
## ARM9 200 MHz as function of cache use

<table>
<thead>
<tr>
<th>cache setting</th>
<th>$t_{\text{context switch}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>From cache</td>
<td>2 $\mu$s</td>
</tr>
<tr>
<td>After cache flush</td>
<td>10 $\mu$s</td>
</tr>
<tr>
<td>Cache disabled</td>
<td>50 $\mu$s</td>
</tr>
</tbody>
</table>
### Context Switch Overhead

The context switch time \( t_{\text{overhead}} \) is given by:

\[
t_{\text{overhead}} = n_{\text{context switch}} \times t_{\text{context switch}}
\]

<table>
<thead>
<tr>
<th>( n_{\text{context switch}} ) (s(^{-1}))</th>
<th>( t_{\text{context switch}} )</th>
<th>CPU load overhead</th>
<th>( t_{\text{overhead}} )</th>
<th>CPU load overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>5ms</td>
<td>0.5%</td>
<td>1ms</td>
<td>0.1%</td>
</tr>
<tr>
<td>5000</td>
<td>50ms</td>
<td>5%</td>
<td>10ms</td>
<td>1%</td>
</tr>
<tr>
<td>50000</td>
<td>500ms</td>
<td>50%</td>
<td>100ms</td>
<td>10%</td>
</tr>
</tbody>
</table>
system performance = f( applications, services, operating system, hardware, tools, what is used?, how often?, how much does it cost?)
Why Quantified Insight in System Design is Required.

version: 0.2
March 6, 2013
QSDpresentationLogoMulti

example of conventional engineering: control engineering

performance engineering: example of quantified SW design

where and how do control and SW engineering meet?

reflection
Impact of Timing on Control Performance

$1/f_{\text{control}}$

measure measure measure

actuate actuate actuate

t_{\text{latency}}

Performance \text{control} = f(f_{\text{control}}, t_{\text{latency}})

0^{e} \text{ order}

$1/e \text{ order}$

impact of jitter on Performance \text{control}

Why Quantified Insight in System Design is Required.
Mutual Impact of SW and Control Design

- **SW design**
  - control algorithm
  - control implementation
  - execution architecture
  - concurrency implementation

- **Control Design**
  - Performance
    - bandwidth
    - type of controller
    - $f_{\text{control}}$
    - $t_{\text{latency}}$
    - stochastic jitter
    - systematic jitter
    - tracking
    - disturbances
    - stability
Impact of digital HW on SW and Control

Why Quantified Insight in System Design is Required.
Why Quantified Insight in System Design is Required.

34

Gerrit Muller
SW design = Multi- Multi- Disciplinary

- control
- algorithm
- control implementation
- execution architecture
- concurrency implementation

Control Design
Performance control
- bandwidth
- type of controller
- \( f_{\text{control}} \)
- \( t_{\text{latency}} \)
- stochastic jitter
- systematic jitter

Physical Design
- mechanical, physics

OS + MW + tools
- digital HW

application
- business
- process
- organization

Why Quantified Insight in System Design is Required.

version: 0.2
March 6, 2013
QSDcontextOfSW

Gerrit Muller
Why Quantified Insight in System Design is Required.

Gerrit Muller
After this slide some more quantification examples and issues are shown
Examples of Quantification; Electronic Patient Record

Customer objectives
figures of merit
patients/physician
physician income
success rate
failure rate
integral cost

Application
typical quantities
# patients
# physicians
# exams/day
# exams/patient
# information/patient

Functional
critical specs
productivity
response time
capacity

Conceptual
working ranges
# transactions
# queries
peak&average

Realization
critical numbers
network speed
CPU speed
memory size
power consumption
query duration
transaction overhead

internal Operational view

market size
market share
growth rate
product life cycle
business model
market segments
maintenance effort
update frequency
service crew
# suppliers
partners
competitors
effort
cost
time
project size
# engineers/discipline
# teams
<table>
<thead>
<tr>
<th>Analysis Type</th>
<th>Considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements Analysis</td>
<td>paradigm boundaries, application relevance, design sensitivity</td>
</tr>
<tr>
<td>Ranges and Relations</td>
<td>typical, best, worst case, dependencies</td>
</tr>
<tr>
<td>Variation Analysis</td>
<td>random vs systematic, types of systematic variation, time-base, rate of change</td>
</tr>
<tr>
<td>Propagation Analysis</td>
<td>amplification or dimming</td>
</tr>
<tr>
<td>Evolution</td>
<td>application, business evolution, technology evolution, scaling, scaling boundaries</td>
</tr>
</tbody>
</table>
Example UI paradigms for Pictorial Index

- Show all
- Add scrolling
- Add hierarchy
- Add meta-information
- Directory
- Meta-information
- Search

Why Quantified Insight in System Design is Required.
Gerrit Muller

version: 0.2
March 6, 2013
QSDpictorialIndex
Example of Combining Heterogeneous Quantifications

- **fields/wafer**
  - best case: 80
  - typical case: 120
  - worst case: 500

- **motion overhead**

- **t_{context switch}**
  - best case: 2 us
  - typical case: 10 us
  - worst case: 50 us

- **throughput**

- **cache settings accesses pollution**

- Why Quantified Insight in System Design is Required.

* $67.2 billion: FBI estimate of what U.S. businesses lose annually because of computer-related crimes.
* $8 billion: Consumer Reports estimate of what U.S. consumers lost the past two years because of viruses, spyware and Internet scams.
* 93.8 million: Privacy Rights Clearinghouse's count of personal records reported lost or stolen since February 2005.
* 26,150: The Anti-Phishing Working Group's count of unique variations of phishing scams reported in August 2006.

Typical costs of goods and services in forums:

* $1,000 to $5,000: Trojan program that can transfer funds between online accounts.
* $500: Credit card number with PIN.
* $80 to $300: Change of billing data, including account number, billing address, Social Security number, home address and birth date.
* $150: Driver's license.
* $150: Birth certificate.
* $100: Social Security card.
* $7 to $25: Credit card number with security code and expiration date.
* $7: PayPal account log-on and password.
* 4% to 8% of the deal price: Fee to have an escrow agent close a complex transaction.
* Free: Access to a service that gives details of the issuing bank for any credit card number.

1 -- Representative asking prices found recently on cybercrime forums

Source: USA TODAY research
referenced by http://groups.google.co.in/group/control-computer-crimes/