Abstract
High level models are simple models with the primary goal to support understanding, analysis, communication and decision making. The models have different complementary representations and formats, e.g. visual diagrams, mathematical formulas, and quantitative information and graphs.

The models are made at different levels to guide software design choices: enterprise level, specification level, and design.
Figure Of Contents™

- reflection on modeling
  - approach
  - complexity
  - high-level

- illustration by case
  - technical
  - context

- problem statement
- conclusions
Ubiquitous Information and Services

consumers

IT infrastructure

information and services anywhere, anytime

IT companies

businesses
government...

High Level Modeling to Support Software Design

version: 0
March 6, 2013
HLMSDubiquitous
But, Horrendous Failure Rate

Slow response, outages, human-less helpdesks, silly excuses (the computer could not...), identity-theft, lost privacy

Budget overruns, total failures

Late delivery of new products, poor scaling of new services, interference of features,

Information and services anywhere, anytime

IT infrastructure

IT companies

Consumers

Businesses

Government

...
Typical Architecture Levels in IT

- Enterprise context
- Enterprise
- System specification
- System design
- Sub-system
- Sub-system
- Sub-system
- Local SW choice
- Enterprise architecture
- Information architecture
- System architecture
High Level Modeling to Support Software Design

Gerrit Muller

version: 0
March 6, 2013
HLMSDlogoApproach
Simplified Framework for Modeling

- **business:** profit, etc.
- operational costs
- stakeholder benefits
- workload
- risks

- **key performance:** throughput, response
- reliability
- availability
- scalability
- ...

- **(emerging?) properties:** resource utilization
- load
- latency, throughput
- quality, accuracy
- ...

**usage context**

- enterprise & users
- requirements
- black box view

**system**

- design
- realization
- technology

**life cycle context**

**and their mutual relations**
Recommendations for Modeling

**objectives**
- support communication
- facilitate reasoning
- support decision making
- create understanding
- maintain insight
- overview

**principles**
- use feedback
- work incremental
- work evolutionary
- be explicit
- make issues tangible

**recommendations**
- Time-box
- Iterate
- Quantify early
- Measure and validate
- Multiple levels of abstraction
- (Simple) mathematical models
- Analysis of accuracy and credibility
- Multi-view
- System and its context
- Visualize

**help to achieve**

**translate into**

**translate into**
High Level Modeling to Support Software Design

Gerrit Muller

version: 0
March 6, 2013
HLMSDlogoTechnical

reflection on modeling
- approach
- complexity
- high-level

illustration by case
- technical
- context

problem statement
conclusions
Example Web Shop

**consumer**
- browse products
- order
- pay
- track

**network**
- exhibit products
- sales & order intake
- order handling
- stock handling
- financial bookkeeping

**enterprise**
- logistics
- finance
- product management
- customer management

**screen**
- client

**web server**
- data base server
- product descriptions
- logistics ERP
- financial
- customer relations

High Level Modeling to Support Software Design
10    Gerrit Muller
Web Shop: NFR’s, Properties and Critical Technologies

**NFR’s:**
- performance browsing
- initial cost
- running costs
- reliability/availability
- scalability order rate
- maintainability
  - effort product changes
  - effort staff changes
- security

**(emerging?) properties**:
- resource utilization
  - server load, capacity
  - memory load, capacity
- response latency
- redundancy
- order throughput
- product data quality
- product definition flow
- staff definition flow
- security design
  - compartimentalization
  - authentication
  - encryption

**Critical technologies**:
- caching
- load balancing
- pipelining
- virtual memory
- memory management
- data base transactions
- XML for customization
- and configuration
- firewalls
- virtual networks

...
Purpose of Picture Cache Model in Web Shop Context

- **Response time**: Required server capacity
  - **Network**: Exhibit products
  - **Web server**: Browse products
  - **Client**: Browse products

- **Server capacity**: Required server capacity
  - **Database server**: Product descriptions
  - **Data base server**: Product descriptions
  - **ERP**: Logistics
  - **Financial**: Financial
  - **Customer relations**: Customer relations
High Level Modeling to Support Software Design

reflection on modeling

problem statement

approach

complexity

high-level

conclusions

illustration by case

technical

context
Level of Abstraction Single System

- Static system definition
- Monodisciplinary
- Number of details

- Multidisciplinary design
- System requirements

- High Level Modeling to Support Software Design

version: 0
March 6, 2013
RAPpyramid
From system to Product Family or Portfolio

![Diagram showing the increase in number of details from system to portfolio. The diagram includes logarithmic scales from $10^0$ to $10^9$ for both the system and portfolio levels.](image)
High Level Modeling to Support Software Design

16  Gerrit Muller

version: 0
March 6, 2013
RAPdiabolo
High Level Modeling to Support Software Design

Gerrit Muller

version: 0
March 6, 2013
HLMSDiabolo
Data Sources of Web Server

- **Content Preparation**
- **Content Provider**
- **Data Quality?**
- **Content**
- **Client**
- **Web Server**
- **Shop Configuration**
  - e.g., staff, roles
- **System Configuration**
  - e.g., resource allocation
new books per year

<table>
<thead>
<tr>
<th>Country</th>
<th>2005 Sales</th>
<th>1996 Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>UK (1)</td>
<td>206k</td>
<td>107k</td>
</tr>
<tr>
<td>USA (2)</td>
<td>172k</td>
<td>68k</td>
</tr>
<tr>
<td>China (3)</td>
<td>107k</td>
<td>101k</td>
</tr>
<tr>
<td>India (21)</td>
<td>12k</td>
<td>12k</td>
</tr>
</tbody>
</table>

source: http://en.wikipedia.org/wiki/Books_published_per_country_per_year

product portfolio characteristics
selection depends on business
life cycle changes determined by business characteristics

source: http://en.wikipedia.org/wiki/Long_tail
### internet: broadband penetration

<table>
<thead>
<tr>
<th></th>
<th>Q1 '04</th>
<th>Q2 '04</th>
<th>growth in Q2 '04</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia Pacific total</td>
<td>48M</td>
<td>54M</td>
<td>12.8%</td>
</tr>
<tr>
<td>China</td>
<td>15M</td>
<td>19M</td>
<td>26.1%</td>
</tr>
<tr>
<td>India</td>
<td>87k</td>
<td>189k</td>
<td>116.8%</td>
</tr>
</tbody>
</table>

What is the expected growth of # customers?
What is the impact on system and infrastructure?
What is the impact on CRM (Customer Relation Management)?
What is the impact on customer, sales support staff?

How much time/effort is needed for content updates?
How much staff is needed?
What is the impact of errors in content updates?
How many errors can be expected?
What is the impact of content updates on server loads?
Web Shop Content Change Effort

Effort on content changes can be estimated as:

$$\text{effort}_{\text{changes}} = n_{\text{changes}} \times (t_{\text{prepare}} + t_{\text{verify}}) + t_{\text{commit}}$$

$$\#fte = \frac{\text{effort}_{\text{changes}}}{\text{hours per day}}$$

<table>
<thead>
<tr>
<th>$n_{\text{changes}}$ per day</th>
<th>10</th>
<th>100</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>effort $\text{changes}$</td>
<td>1 uur</td>
<td>10 uur</td>
<td>100 uur</td>
</tr>
<tr>
<td>$#fte$</td>
<td>0.1</td>
<td>1</td>
<td>12</td>
</tr>
</tbody>
</table>

with:
- $t_{\text{prepare}} = 4$ min
- $t_{\text{verify}} = 2$ min
- $t_{\text{commit}} = 1$ min

hours per day = 8 hours
Example of Client Level Changes

Up-to-date information:
Bestsellers
What Other Customers Are Looking At Right Now
catalogue entries
main access through search
personalization
other advertisements
styling: frequently updated, fashion!
snapshot of www.amazon.com

standard boilerplate

High Level Modeling to Support Software Design

version: 0
March 6, 2013
MALCmoreChanges
Web Shop Security and Changes

What is the security model?
What is the impact on server loads?
What is the impact on staffing?
What is the impact of changes in staff?
What is the impact of changes on security?

High Level Modeling to Support Software Design
25 Gerrit Muller
new faults = average fault density * #changes

#errors = \sum_{\text{faults}} f( \text{severity}, \text{hit probability}, \text{detection probability})

<table>
<thead>
<tr>
<th>severity</th>
<th>hit probability</th>
<th>detection probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jansen iso Janssen</td>
<td>low</td>
<td>high</td>
</tr>
<tr>
<td>operator iso sales repr</td>
<td>high</td>
<td>high</td>
</tr>
</tbody>
</table>
High Level Models Support Communication

- Number of involved people
- Time per iteration cycle

Model based architecting

- Fast but intangible
- Tangible

In the architect's head

Graph showing the relationship between the number of involved people and the time per iteration cycle, illustrating the concept of model-based architecting, which is fast but intangible.
Model Life Cycle

understanding → exploration → optimization → verification

- Try out models: most try out models never leave the desk or computer of the architect!
- Simple and small models: many small and simple models are used only once; some are re-used in next projects
- Substantial models: substantial models capture core domain knowledge; they evolve often from project to project.

Substantial models are used and maintained, while simple and small models are often abandoned or archived.

High Level Modeling to Support Software Design
Gerrit Muller

version: 0
March 6, 2013
MAREmodelLifeCycle
Example Model Life Cycle

understanding → exploration → optimization → verification

try out models
load/cost → function mix → load/cost peak impact
simple and small models
customer global distribution → integral load model → webshop benchmark suite
substantial models (IP assets)
global customer demographics → load/stress test suite → web server performance

High Level Modeling to Support Software Design
version: 0
March 6, 2013
Gerrit Muller
lack of integral understanding
of software in human and business context
causes horrendous failure rate of IT projects and systems

simple high-level models
create understanding across specialties
and stimulate fact based decision making