Abstract
This presentation uses a web shop service as example system to construct a system model. The caching of pictures of the products in the shop is modeled to analyze performance, robustness, scalability and reliability of the system.

Hit rate is context dependent. Life cycle changes or peak loads may degrade hit rate.
What to model of the system

Stepwise approach to system modeling

Non Functional requirements (NFR), System Properties and Critical Technologies

Examples of web shop case
What to Model in System Context?

**usage context**
- enterprise & users

**system**
- (emerging?) properties:
  - resource utilization
  - load
  - latency, throughput
  - quality, accuracy
  - sensitivity (changes, inputs)

- critical technologies:
  - caching
  - load balancing
  - firewalls
  - virtual networks
  - XML for customization and configuration

**NFR's:**
- performance
- reliability
- availability
- scalability
- maintainability

**life cycle context**
- creation
- life cycle business
1. determine relevant Non Functional Requirements (NFR's)

2. determine relevant system design properties

3. determine critical technologies

4. relate NFR's to properties to critical technologies

5. rank the relations in relevancy and criticality

6. model relations with a high score
**NFR’s:**
- performance browsing
- initial cost
- running costs
- reliability/availability
- scalability order rate
- maintainability
  - effort product changes
  - effort staff changes
- security

**System: (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
  - compartmentalization
  - authentication
  - encryption

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

**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
  - compartmentalization
  - authentication
  - encryption

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

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

(system)
NFR’s:
resource utilization
server load, capacity
memory load, capacity
response latency
redundancy
order throughput

(emerging?) properties:

critical technologies:
caching
load balancing
pipelining
virtual memory
memory management

security design:
compartimentalization
authentication
encryption

security design:
firewalls
virtual networks

ranking will be discussed in
Modeling and Analysis: Reasoning
Purpose of Picture Cache Model in Web Shop Context

Modeling and Analysis: System Model

version: 0.4
March 6, 2013
MASMwebShopPictureCache

Gerrit Muller
Zero Order Load Model

**zero order web server load model**

\[
\text{Load} = n_a \times t_a
\]

- \(n_a\) = total requests
- \(t_a\) = cost per request
First Order Load Model

**first order web server load model**

\[
\text{Load} = n_{a,h} \cdot t_h + n_{a,m} \cdot t_m
\]

- \(n_{a,h}\) = accesses with cache hit
- \(n_{a,m}\) = accesses with cache miss
- \(t_h\) = cost of cache hit
- \(t_m\) = cost of cache miss

\[
n_{a,h} = n_a \cdot h
\]

\[
n_{a,m} = n_a \cdot (1-h)
\]

- \(n_a\) = total accesses
- \(h\) = hit rate

\[
\text{Load}(h) = n_a \cdot h \cdot t_h + n_a \cdot (1-h) \cdot t_m = n_a \cdot t_m - n_a \cdot h \cdot (t_m - t_h)
\]
Quantification: From Formulas to Insight

quantified mid office server example

\[ t_h = 0.02 \text{ ms} \]
\[ t_m = 2 \text{ ms} \]
\[ n_a = 1000 \]

Load(h) = 1000 \( \ast \) 2[ms] - 1000 \( \ast \) h \( \ast \) 1.98[ms]

Load(h) = 2000 - 1980 \( \ast \) h [ms]

Modeling and Analysis: System Model

version: 0.4
March 6, 2013
MASMquantified
Hit Rate Considerations

**Quantified Mid Office Server Example**

- \( t_h = 0.02 \text{ ms} \)
- \( t_m = 2 \text{ ms} \)
- \( n_a = 1000 \)

\[
\text{Load}(h) = 1000 \times 2[\text{ms}] - 1000 \times h \times 1.98[\text{ms}]
\]

\[
\text{Load}(h) = 2000 - 1980 \times h[\text{ms}]
\]

Hit rate is context dependent. *Life cycle changes* or *peak loads* may degrade hit rate.

Hit rate of well designed system is ample within working range (e.g. 95%). 0\(^{th}\) order formula is valid:

\[
\text{Load} = 0.12 \times n_a[\text{ms}]
\]
Response Time

1. Human (customer) presses "next".
2. Client requests picture.
3. Web server checks cache.
4. If in cache, retrieves picture.
5. If not in cache, requests picture from data base server.
7. Web server stores picture in cache.
8. Web server transfers picture to client.
10. Client displays picture.

Time in milliseconds in optimal circumstances:
- $t_0$:
- $t_0 + 10$
- $t_0 + 20$
- $t_0 + 30$
- $t_0 + 40$
- $t_0 + 50$
- $t_0 + 60$
- $t_0 + 70$
What Memory Capacity is Required for Picture Transfers?

Modeling and Analysis: System Model
14 Gerrit Muller

version: 0.4
March 6, 2013
MASMwebShopPictureFlow
Process View of Picture Flow in Web Server

- **One copy per process**
  - mid office server
  - web server
  - picture cache server
  - back office access

- **Multiple copies per process**
  - mid office server
  - web server
  - picture cache server
  - back office access

- **Multiple copies per process and thread**
  - mid office server
  - web server
  - picture cache server
  - back office access

\[ \text{Modeling and Analysis: System Model} \]

15 Gerrit Muller

version: 0.4
March 6, 2013
MASMmidOfficeProcesses
picture memory =

\[ 3 * n * s + 5 * m * s + c * s + 3 * k * s \]

where

- \( n \) = number of database access threads
- \( m \) = number of picture cache threads
- \( k \) = number of web server threads
- \( s \) = picture size in bytes
- \( c \) = in-memory cache capacity in number of pictures
### Web Server Memory Capacity

**Picture Memory Equation:**

\[ \text{picture memory} = 3 \, n \, s + 5 \, m \, s + c \, s + 3 \, k \, s \]

Where:
- \( n \) = number of back office access threads
- \( m \) = number of picture cache threads
- \( k \) = number of web server threads
- \( s \) = picture size in bytes
- \( c \) = in-memory cache capacity in \# pictures

**Use Cases and Memory Use:**

<table>
<thead>
<tr>
<th>Use Case</th>
<th>n</th>
<th>m</th>
<th>k</th>
<th>s</th>
<th>c</th>
<th>MB</th>
<th>Storage Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small Shop</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>100</td>
<td>10</td>
<td>1.5</td>
<td>L3</td>
</tr>
<tr>
<td>Highly Concurrent</td>
<td>2</td>
<td>4</td>
<td>100</td>
<td>100</td>
<td>20</td>
<td>5.3</td>
<td>Main</td>
</tr>
<tr>
<td>Large Pictures</td>
<td>2</td>
<td>4</td>
<td>1000</td>
<td>100</td>
<td>100</td>
<td>296</td>
<td>Main+Disk</td>
</tr>
<tr>
<td>Many Pictures</td>
<td>2</td>
<td>4</td>
<td>10</td>
<td>100</td>
<td>100,000</td>
<td>9,540</td>
<td>Main+Disk</td>
</tr>
<tr>
<td>All at Once</td>
<td>2</td>
<td>4</td>
<td>1000</td>
<td>100</td>
<td>100,000</td>
<td>98,234</td>
<td>Disk</td>
</tr>
</tbody>
</table>

**Processor Caches:**

- L1: \( \text{kB} \)
- L2: \( \text{MB} \)
- L3: \( \text{GB} \)
- Main Memory
- Disk

**Memory Use:**
- Product browsing only
- Pictures only
- Single server

**Question:** What is the performance impact of memory use on other processing?
We Have only Modeled a Small Part of the System...

| function       | browse/exhibit products | sales, order intake, payments
|                |                        | track, order handling
|                |                        | stock handling
|                |                        | financial bookkeeping
|                |                        | customer relation management
|                |                        | update catalogue
|                |                        | advertize
|                |                        | after sales support

| data          | picture                | structured (product attributes, logistics, ...)
|              |                        | program code

| aspect        | server memory use      | network use
|              | response time          | reliability
|              | server load            | any resource, any NFR

aspect(d, f)  =  \sum_{d = all data} \sum_{f = all functions} aspect(d, f)

ignoring other dimensions such as applications, users, circumstances
static
mostly assumptions and coarse estimates
some insight in:
what are key design issues
what are relevant use case areas
Refinement After Context Modeling

usage context

enterprise & users

NFR's:
- performance
- reliability
- availability
- scalability
- maintainability
...

(system)

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

critical technologies:
- caching
- load balancing
- firewalls
- virtual networks
- XML for customization and configuration
...

life cycle context

life cycle business

creation
Conclusions

Non Functional Requirements are the starting point for system modeling

Focus on highest ranking relations between NFR’s and critical technologies

Make simple mathematical models

Evaluate quantified instantiations

Techniques, Models, Heuristics of this module

Non functional requirements

System properties

Critical technologies

Graph of relations