Abstract

This module addresses the integration of small or partial models into bigger models. We also discuss how multiple models are used and how to reason using multiple models.
Module Modeling and Analysis: Integration and Reasoning

Where are we in the Course?

- facts from research
- measurements
- assumptions

Modeling and Analysis: Integration and Reasoning

- uncertainties
- working range
- accuracy
- credibility
- risk
- time, cost, effort
- profit margin

- specification
- verification
- decisions

- project

- reasoning

- modeling
- analysis

- system
- usage context

- enterprise & users
- requirements black box view
- design realization technology
- creation life cycle business
- life cycle context

- version: 0.3
- September 9, 2018
Abstract

Models are made to facilitate decision making. These decisions range from business decisions, such as Service Level Agreements, to requirements, and to detailed design decisions. The space of decisions is huge and heterogeneous. The proposed modeling approach is to use multiple small and simple models. In this paper we discuss how to reason by means of multiple models.
From chaos to order: inputs, assumptions, models and decisions
Reasoning approach: stepwise top-down and bottom-up
Life cycles of models in relation to project life cycle
Purpose of Modeling

How to use multiple models to facilitate decisions?
How to get from many fragments to integral insight?
How many models do we need?
At what quality and complexity levels?

Key terms:
- facts from investigation
- measurements
- assumptions
- uncertainties
- unknowns
- errors
- modeling
- analysis
- results
- decision making
- specification
- verification
- decisions
- accuracy
- working range
- credibility
- risk
- customer satisfaction
- time, cost, effort
- profit margin
Example Graph for Web Shop

**usage context**
- enterprise & users
  - customer interest
  - customer behavior
  - personnel
  - financial
  - salary
  - workflow
  - service cost

**system**
- black box view
  - load
  - response time
  - throughput
  - information
  - elapsed time
  - budget

- design
  - transactions
  - CPU load
  - network load
  - storage capacity
  - CPU budget
  - memory budget
  - overhead
  - picture cache
  - access time

**life cycle context**
- running cost
- initial cost
- maintenance effort
- changes
- service
- cost

**SLA**

**legend**
- a: assumption
- i: input e.g. measurement
- d: decision
- m: model

**version: 0.6**
September 9, 2018
MAREgraphWebShop
Relations: Decisions, Models, Inputs and Assumptions

Modeling and Analysis: Reasoning
Gerrit Muller
version: 0.6
September 9, 2018
Reasoning Approach

1. Explore usage context, life cycle context and system

2. Determine main Threads-of-Reasoning

3. Make main Threads-of-Reasoning SMART

4. Identify "hottest" issues

5. Model hottest, non-obvious, issues

6. Capture overview, results and decisions

7. Iterate and validate

all steps time-boxed between 1 hour and a few days
early in project later in project

- t2a. "Play" with models
- t2b. Investigate facts
- t2c. Identify assumptions
- b3. Model significant, non-obvious, issues

Modeling and Analysis: Reasoning
version: 0.6
September 9, 2018
MAREmethod

Gerrit Muller
1. Explore usage context, life cycle context and system

Populate with "known" facts, numbers, issues from preceding projects, available work products and stakeholders

**usage context**
- products & attributes
- customers
- personnel
- order rate

**system**
- functions
- interfaces
- price
- performance

**life cycle context**
- SW/HW platform(s)
- resources
- cost
- transaction load/latency

- infrastructure
- margin services
- maintenance projection
t2. Determine main Threads-of-Reasoning

Architecting and System Design

e.g. http://www.gaudisite.nl/ModuleTORSlides.pdf
t3. SMART’en Thread-of-Reasoning

Quantify in terms of Key Performance Indicators, Key Performance Measures, Critical Resources

- market dynamics
- service costs

usage context
- products & attributes
- resources
- order rate
- personnel

functions
- interfaces
- price
- performance

SW/HW platform(s)
- resources
- cost
- transaction load
- latency

life cycle context
- infrastructure
- energy services
- maintenance projections

JFMAMJJASOND
- time

system
- order rate
- server cost
- market volume

best case
worst case

version: 0.6
September 9, 2018
MAREthreadSMART
Intermezzo: the acronym SMART

<table>
<thead>
<tr>
<th>Specific</th>
<th>quantified</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measurable</td>
<td>verifiable</td>
</tr>
<tr>
<td>Assignable</td>
<td>(Achievable, Attainable, Action oriented, Acceptable, Agreed-upon, Accountable)</td>
</tr>
<tr>
<td>Realistic</td>
<td>(Relevant, Result-Oriented)</td>
</tr>
<tr>
<td>Time-related</td>
<td>(Timely, Time-bound, Tangible, Traceable)</td>
</tr>
</tbody>
</table>

acronym consensus

variation of meaning
t4: Identify Hottest

**t4. Identify "hottest" issues**

Assess explored landscape:
- Highest (perceived) risk
- Most important/valuable
- Most discussed
- Historic evidence
- Urgency

Rank issues according to aggregated assessment

<table>
<thead>
<tr>
<th>Risk</th>
<th>Value</th>
<th>Discussion</th>
<th>History</th>
<th>Urgency</th>
<th>Ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>server cost</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>order rate</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>transactions</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>response time</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>availability</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>network bandwidth</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>storage capacity</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
t5. Model hottest, non-obvious, issues

- **Best case**
  - Server capacity

- **Worst case**
  - Market volume
  - Order rate
  - Time

**Non-obvious:**
- Desired lead time 2 months
- Versus actual lead time 6 months

---

Modeling and Analysis: Reasoning
15 Gerrit Muller

version: 0.6
September 9, 2018
MAREmodelHottest
From *top-down* to *bottom-up*

1. Explore usage context, life cycle context and system

   **top-down**

   t2. Determine main Threads-of-Reasoning

   t3. Make main Threads-of-Reasoning SMART

   t4. Identify "hottest" issues

   t5. Model hottest, non-obvious, issues

   **bottom-up**

   b2a. "Play" with models

   b2b. Investigate facts

   b2c. Identify assumptions

   b3. Model significant, non-obvious, issues

   **learn**

6. Capture overview, results and decisions

7. Iterate and validate

*all steps time-boxed between 1 hour and a few days*

early in project later in project
### Bottom-up

<table>
<thead>
<tr>
<th>b2a. &quot;Play&quot; with models</th>
<th>b2b. Investigate facts</th>
<th>b2c. Identify assumptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>vary inputs</td>
<td>market research</td>
<td>What is the basis for</td>
</tr>
<tr>
<td>vary model structure</td>
<td>measurements preceeding systems</td>
<td>model structure, design decision, specification, quantification et cetera?</td>
</tr>
<tr>
<td>to understand <em>model applicability, design quality and specification feasibility</em></td>
<td>micro benchmarks literature, supplier info</td>
<td><em>Most assumptions are implicit and hidden!</em></td>
</tr>
</tbody>
</table>

\[
n_{\text{CPU}} = \frac{t_{\text{required total}}}{t_1 \text{ CPU}}
\]

\[
t_{\text{required total}} = n_{\text{transactions}} \times t_1 \text{ transaction} + t_{\text{other}}
\]

\[
t_1 \text{ transaction} = 1 \text{ ms (on 1 CPU)}
\]

http://www.tpc.org/tpcc/results/tpcc_perf_results.asp

---

**IBM System p5 595**

TPC-C Throughput 4,033,378

- Total # of Processors: 32
- Total # of Cores: 64

\[
\frac{1}{t_1 \text{ transaction}} = 4 \times 10^6 / 60 / 64 \text{ min to sec / # cores}
\]

\[
t_1 \text{ transaction} \sim 1 \text{ ms}
\]

---

server load dominated by transactions

transaction load scales linear

TPC-C is representative

what is the effect of other TPC-C workload?
Bottom-up, more detailed steps

| Make a list of technologies, components and resources to be used |
| transactions, data base engine, memory, disk |

| Make a list of important qualities |
| performance, reliability, security, maintainability |

| Make a characterization matrix of technologies, components and resources versus qualities |
| 1..5 scale, 1 = low risk, 5 = high risk et cetera |

| Perform step 2abc on most critical |
| class 4 and 5 risks |
b3. Model significant, non-obvious, issues

for example, memory use in server(s) for picture transfers and buffering

The performance impact of memory use on other processing can be calculated using the following formula:

\[
picture memory = 3 \times n \times s + 5 \times m \times s + c \times s + 3 \times k \times s
\]

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

\[
\begin{align*}
n & \quad m & \quad k & \quad s & \quad c & \quad MB \\
1 & \quad 1 & \quad 1.E+05 & \quad 10 & \quad 1.5 \\
2 & \quad 4 & \quad 10.E+05 & \quad 20 & \quad 5.3 \\
2 & \quad 4 & \quad 1000.E+05 & \quad 100 & \quad 296.2 \\
2 & \quad 4 & \quad 1000.E+06 & \quad 100 & \quad 2962.1 \\
\end{align*}
\]

What is the performance impact of memory use on other processing?

**Diagram**:
- **Mid Office Server**
- **Web Server**
- **Picture Cache Server**
- **Back Office Access**

**Legend**:
- One copy per process
- Multiple copies per process
- Multiple copies per process and thread

**Notes**:
- Memory use for product browsing only pictures on single server
- Several servers with different memory configurations

**Questions**:
- How does memory use impact performance?
- What is the role of each server in the system?
Learning Concurrent Bottom-up and Top-down

top-down: what is hidden in details?
top-down: do we address the relevant decomposition?
bottom-up: do we address relevant details?
bottom-up: what details have significant impact?

order rate
(and evolution over time)
is highly speculative

transactions
dominate
server load & cost

needs
input
induces
risk
Example top-down and bottom-up

top-down:
what is impact of
catalogue size and changes?

bottom-up:
what is relevant concurrency (k), cache size (c),
or picture size (s)?

<table>
<thead>
<tr>
<th>country</th>
<th>L1</th>
<th>L2</th>
<th>L3</th>
<th>MB</th>
<th>storage type</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA (2)</td>
<td>68k (1996)</td>
<td>4</td>
<td>10</td>
<td>100,000</td>
<td>20</td>
</tr>
<tr>
<td>China (3)</td>
<td>101k (1994)</td>
<td>4</td>
<td>1000</td>
<td>100,000</td>
<td>100</td>
</tr>
<tr>
<td>India (21)</td>
<td>12k (1996)</td>
<td>2</td>
<td>1000</td>
<td>1,000,000</td>
<td>100</td>
</tr>
</tbody>
</table>

Memory use
- Product browsing only pictures only single server
- Memory use:
  - L1 capacity
  - L2
  - L3
  - Main memory
  - Disk

What is the performance impact of memory use on other processing?

- New books per year
  - Source: http://en.wikipedia.org/wiki/Books_published_per_country_per_year

- Amazon "long tail"
  - Source: http://en.wikipedia.org/wiki/Long_tail

- WH Smith

- Item sales

- Version: 0.6
  - September 9, 2018
  - MARElearnExample
6. Capture overview, results and decisions

- Definition
- Execution
- Deployment

Major milestone

Transfer of relevant insights

Re-usable assets from previous projects

Re-usable assets to next projects

Customer business

Proposal

Specification

Design

Realization

Configuration

Consolidation baseline
7. Iterate and Validate

stakeholders

questions
results
data
experiments
measurements
data

systems, models, components

1         iteration 2         iteration 3         iteration 4         time

1 2 3 t 4 5 6 7 t 2 3 t 4 t 5 6 7 t 2 3 t 4 t 5 6 7
Focus is Shifting during Project

- **usage context**
  - conception & feasibility
  - requirements elicitation
  - design and realization
  - validation

- **life cycle context**
  - conception & feasibility
  - requirements elicitation
  - design and realization
  - validation

- **system**
  - conception & feasibility
  - requirements elicitation
  - design and realization
  - validation
Models Support Communication

- Number of involved people vs. time per iteration cycle
- In the architect's head: model based architecting
- Fast but intangible
- Tangible

Graph shows the relationship between the number of involved people and the time per iteration cycle. The graph includes a curve indicating a fast but intangible process, leading to tangible outcomes in model-based architecting.
Frequency of Assumptions, Decisions and Modeling

- Implicit (trivial?)
- Explicit
- Try-outs
- Very simple
- Small
- Key
- Substantial

Legend:
- Assumption
- Input e.g. measurement
- Decision
- Model

Graph showing the frequency of assumptions, decisions, and modeling with various categories ranging from trivial to substantial.
Life Cycle of Models

- Understanding
- Exploration
- Optimization
- Verification

- Try out models
  - Most try out models never leave the desk or computer of the architect!
  - Many small and simple models are used only once; some are re-used in next projects

- Simple and small models
  - Re-used in next project
  - Archived
  - Not maintained

- Substantial models
  - Re-used in next project
  - Re-used in next project
  - Re-used in next project
  - Archived
  - Not maintained
  - Archived
  - Not maintained

Substantial models capture core domain know how; they evolve often from project to project. Creation and evolution of intellectual property assets
Examples of Life Cycle of Models

- **Understanding**
  - Try out models
  - Simple and small models
  - Substantial models (IP assets)

- **Exploration**
  - Function mix
  - Customer global distribution
  - Global customer demographics
  - Integral load model
  - Load/stress test suite

- **Optimization**
  - Load/cost
  - Peak impact
  - Web server performance

- **Verification**
  - Webshop benchmark suite
Conclusions

Top-down and bottom-up provide complementary insights

Key words for selection: hottest, non-obvious, significant, relevant

Multiple small models are used in combination

Some models evolve from very simple to more substantial

Techniques, Models, Heuristics of this module

Threads-of-reasoning

SMART

Key Performance Indicators, Key Performance Measures, Critical Resources

Ranking matrices
Abstract

More substantial models are created step by step. We will discuss the order of creation and modularity considerations. The modules have to be integrated into the desired substantial model.
Example of (Partial) Flow Simulator

Modeling and Analysis: Modularity and Integration
Gerrit Muller
Example of Incremental Model Creation
## Approach for Incremental Model Creation

<table>
<thead>
<tr>
<th>Step</th>
<th>Recommendation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Start with the hottest issue&lt;br&gt; <strong>what creates the most discussion or uncertainty?</strong></td>
</tr>
<tr>
<td>2.</td>
<td>Ensure immediate feedback&lt;br&gt; <strong>does this model help to answer the questions that we have?</strong></td>
</tr>
<tr>
<td>3.</td>
<td>Keep flexible decoupling point&lt;br&gt; e.g. human readable/editable files</td>
</tr>
<tr>
<td>4.</td>
<td>Extend model only for a good purpose&lt;br&gt; don't integrate models because it <em>can</em> be done</td>
</tr>
<tr>
<td>5.</td>
<td>Create effective visual outputs&lt;br&gt; simple animations, graphs, tables, ...</td>
</tr>
<tr>
<td>6.</td>
<td>Refactor regularly&lt;br&gt; based on increasing insight, feedback and purpose</td>
</tr>
</tbody>
</table>
Attention Points for Every Integration Step

Does the output of the integrated model match your expectation?
Can you explain the model behavior?
Can you explain the variation of the output?

\[ \text{load} \]
\[ \text{time} \]
\[ \text{time behavior} \]

\[ \text{value} \]
\[ \text{variation} \]

or

\[ ? \]