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.
Where are we in the Course?

- facts from research
- measurements
- assumptions
- uncertainties
- unknowns
- errors
- assumptions
- modeling
- analysis
- accuracy
- working range
- credibility
- project
- specification
- verification
- risk
- customer satisfaction
- time, cost, effort
- profit margin

Module Modeling and Analysis: Integration and Reasoning
©2006, Embedded Systems Institute

version: 0.3
July 24, 2014
MMAREposition

©2006, Gerrit Muller
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.

Distribution
This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

July 24, 2014
status: preliminary
draft
version: 0.6
content

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
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?
Graph of Decisions and Models
Example Graph for Web Shop

usage context
enterprise & users
- customer interest
- market penetration
- #products
- customer behavior
- marginal
- financial
- personnel
- work flow
- service cost

black box view
- load
- response time
- throughput
- information
- transaction speed
- transaction CPU
- elapsed time budget

system
design
- transactions
- transactions
- CPU load
- network load
- storage capacity
- CPU budget
- memory budget
- overhead

life cycle context
SLA
- running cost
- initial cost
- maintenance effort
- changes

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

Modeling and Analysis: Reasoning
©2006, Embedded Systems Institute
©2006, Gerrit Muller
version: 0.6
July 24, 2014
MAREgraphWebShop
Relations: Decisions, Models, Inputs and Assumptions

Diagram:
- **i**: Represents input e.g., measurement
- **d**: Represents decision
- **m**: Represents model

Legend:
- **a**: Assumption

Connections:
- Input feeds decision
- Decision triggers model
- Model calibrates input
- Assumption influences decision

Tags:
- Facilitates
- Triggers
- Feeds
- Influenz
Reasoning Approach

1. Explore usage context, life cycle context and system
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

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
1. Explore usage context, life cycle context and system

Populate with "known" facts, numbers, issues from preceding projects, available work products and stakeholders
t2. Determine main Threads-of-Reasoning

Architecting and System Design

usage context
- market dynamics
- service costs
- customers
- order rate
- personnel

functions
- interfaces
- price
- performance

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

infrastructure
- margin services
- maintenance projection

life cycle context

e.g. http://www.gaudisite.nl/ModuleTORSslides.pdf
t3. Make main Threads-of-Reasoning SMART

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

- market dynamics
- service costs

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

- functions
  - interfaces
  - price
  - performance

- system
  - SW/HW platform(s)
  - resources
  - cost
  - transaction/interaction

- life cycle context
  - infrastructure
  - margin services
  - maintenance projection

- time
- market volume
- best case
- worst case

- order rate
- server cost
- 2M$
- 1M$

- system load

version: 0.6
July 24, 2014
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>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Assignable (Achievable, Attainable, Action oriented, Acceptable, Agreed-upon, Accountable)</th>
</tr>
</thead>
</table>

| Realistic (Relevant, Result-Oriented) |

| Time-related (Timely, Time-bound, Tangible, Traceable) |

*variation of meaning*
t4: Identify Hottest

<table>
<thead>
<tr>
<th>assess explored landscape:</th>
<th>1..5 scale, 1 = low risk, 5 = high risk et cetera</th>
</tr>
</thead>
<tbody>
<tr>
<td>highest (perceived) risk</td>
<td>5 = high risk</td>
</tr>
<tr>
<td>most important/valueable</td>
<td></td>
</tr>
<tr>
<td>most discussed</td>
<td></td>
</tr>
<tr>
<td>historic evidence</td>
<td></td>
</tr>
<tr>
<td>urgency</td>
<td></td>
</tr>
</tbody>
</table>

rank issues according to aggregated assessment

<table>
<thead>
<tr>
<th>server cost</th>
<th>2</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>3</th>
</tr>
</thead>
<tbody>
<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>

Modeling and Analysis: Reasoning
©2006, Embedded Systems Institute
©2006, Gerrit Muller

version: 0.6
July 24, 2014
MAREhottest
t5. Model hottest, non-obvious, issues

- **Best case** server capacity vs. **worst case**
- **Non-obvious:** desired lead time 2 months versus actual lead time 6 months
From *top-down* to *bottom-up*

1. Explore usage context, life cycle context and system

   - t2. Determine main Threads-of-Reasoning
   - t3. Make main Threads-of-Reasoning SMART
   - t4. Identify "hottest" issues
   - t5. Model hottest, non-obvious, issues

2. Determine main Threads-of-Reasoning

   - b2a. "Play" with models
   - b2b. Investigate facts
   - b2c. Identify assumptions
   - b3. Model significant, non-obvious, issues

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

version: 0.6  
July 24, 2014  
MAREmethod
### b2abc: 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</td>
<td>model structure, design</td>
</tr>
<tr>
<td>to understand model</td>
<td>preceding systems</td>
<td>decision, specification,</td>
</tr>
<tr>
<td>applicability, design</td>
<td>micro benchmarks</td>
<td>quantification et cetera?</td>
</tr>
<tr>
<td>quality and specification feasibility</td>
<td>literature, supplier info</td>
<td>! Most assumptions are implicit and hidden!</td>
</tr>
</tbody>
</table>

\[ n_{CPU} = \frac{t_{\text{required total}}}{t_1 \text{ CPU}} \]
\[ t_{\text{required total}} = n_{\text{transactions}} \times t_1 \text{ transaction} + t_{\text{other}} \]
\[ t_1 \text{ transaction} = 1 \text{ ms (on 1 CPU)} \]

---

**IBM System p5 595**

TPC-C Throughput: 4,033,378

Total # of Processors: 32

Total # of Cores: 64

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

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

---

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 Issues

b3. Model significant, non-obvious, issues

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

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

where

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

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

memory use
- product browsing only
- pictures only
- single server

What is the performance impact of memory use on other processing?
Learning Concurrent Bottom-up and Top-down

top-down  learn  bottom-up

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*

needs input ---- transactions dominate
input --- induces risk
server load&cost
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)?

new books per year

UK (1) 206k (2005) 107k (1996)
USA(2) 172k (2005) 68k (1996)
China(3) 101k (1994)
India(21) 12k (1996)

memory use
product browsing only
pictures only
single server

memory type
L1 capacity L2 L3 main memory disk

sales
Amazon "long tail"

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

picture memory = 3 * n * s + 5 * m * s + c * s + 3 * k * s

n = # back office access threads
m = # picture cache threads
k = # web server threads
s = picture size in bytes
c = in memory cache capacity in # pictures

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

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

Modeling and Analysis: Reasoning
©2006, Embedded Systems Institute
version: 0.6
July 24, 2014
MARElearnExample
6. Capture overview, results and decisions

- re-usable assets from previous projects
- major milestone
- transfer of relevant insights
- re-usable assets from previous projects
- major milestone
- transfer of relevant insights
- re-usable assets to next projects
- customer business
- proposal
- specification
- design
- realization
- consolidation baseline
- configuration
7. Iterate and Validate

stakeholders

questions
results
data

experiments
measurements
data

systems, models, components

1 iteration 2 iteration 3 iteration 4 — time —
Focus is Shifting during Project

usage context

life cycle context

system

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

- Number of involved people
- Time per iteration cycle
  - 1 minute
  - 1 hour
  - 1 day
  - 1 week
  - 1 month

- Model based architecting
  - Tangible
  - Fast but intangible

- In the architect's head
Frequency of Assumptions, Decisions and Modeling

The diagram illustrates the frequency of assumptions, decisions, and modeling across different scales.

- **Implicit (trivial?)**
  - Frequency: $10^0$
  - Color: Yellow

- **Explicit**
  - Frequency: $10^4$
  - Color: Yellow

- **Try-outs**
  - Frequency: $10^6$
  - Color: Pink

- **Very simple**
  - Frequency: $10^0$
  - Color: Pink

- **Small**
  - Frequency: $10^2$
  - Color: Pink

- **Key**
  - Frequency: $10^4$
  - Color: Yellow

- **Substantial**
  - Frequency: $10^6$
  - Color: Pink

**Legend:**
- A: Assumption
- I: Input, e.g., measurement
- D: Decision
- M: Model

**Note:**
- The chart shows a distribution of assumptions, decisions, and models across different scales.
Life Cycle of Models

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 know how;
they evolve often from project to project.
creation and evolution of intellectual property assets
Examples of Life Cycle of Models

- 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
- Global customer demographics

Substantial models (IP assets)

- Load/stress test suite
- Web server performance
- 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.

Distribution
This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.
Example of (Partial) Flow Simulator

Modeling and Analysis: Modularity and Integration
©2006, Gerrit Muller
Example of Incremental Model Creation
## Approach for Incremental Model Creation

<table>
<thead>
<tr>
<th>Recommendation</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start with the hottest issue</td>
<td>what creates the most discussion or uncertainty?</td>
</tr>
<tr>
<td>Ensure immediate feedback</td>
<td>does this model help to answer the questions that we have?</td>
</tr>
<tr>
<td>Keep flexible decoupling point</td>
<td>e.g. human readable/editable files</td>
</tr>
<tr>
<td>Extend model only for a good purpose</td>
<td>don't integrate models because it can be done</td>
</tr>
<tr>
<td>Create effective visual outputs</td>
<td>simple animations, graphs, tables, ...</td>
</tr>
<tr>
<td>Refactor regularly</td>
<td>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?

![Diagram showing time behavior and variation of value](image)