Abstract
The architect needs an integral view of a system in its context. The level of detail used in the integral view is limited by necessity. This view is based on a much more detailed, but implicit, understanding of system and context. Many experience based methods are used by the architect to reach an explicit understanding of the functionality and performance of the system, black box as well as white box.

The dynamic range of mental architectural activity is explained. The thinking process of the architect is illustrated by means of budgetting, one of the many experience based methods of architects.
Outline of the Presentation

Introduction
  - scope
  - case

Problem
  - complex systems
  - shortage of architects

Solution
  - H2 transfer? H2 research?
  - from implicit experience to explicit know how

Analysis
  - How do architects think?
Embedded Systems; From Small to Large

- chip
- GSM
- television
- printer
- waferstepper
- MRI scanner
- cardio X-ray system
What is a Waferstepper

source
reticle
lens
wafer
From stepping to scanning

**stepper:** static exposure of field

**scanner:** dynamic exposure through slit

- stepper:
  - reticle
  - slit
  - wafer: 250 mm/s

- scanner:
  - expose
  - step
  - expose

Architectural Thinking
6 Gerrit Muller

version: 1.1
March 6, 2013
ASMLstepperVsScanner
Key specifications waferstepper (2001)

- Imaging
  - 130 nm line width
  - Critical dimension: 10 nm
  - Overlay: 45 nm

- Alignment
Moore’s law

line width in nm

1997 1999 2001 2003 2005 2007

250 180 130 100 70 50

Architectural Thinking

version: 1.1
March 6, 2013
ASMLmooreLaw
Embedded systems: software intensive
large development teams
multi-disciplinary

Systems Architecting is an art
Skilled architects are scarce

Failures, delays, non-performance, cost overruns, dissatisfied customers

Eberhardt Rechtin and Mark W. Maier.
The Art of Systems Architecting.
Symptom: Delays appear during Integration

Do you have any design issues for the design meeting?

The default answer is: No.

During integration numerous problems become visible

version: 1.1
March 6, 2013
MSintegration
Exponential Pyramid, from requirement to bolts and nuts

10^0

10^1

10^2

10^3

10^4

10^5

10^6

10^7

Architectural Thinking
version: 1.1
March 6, 2013
IALApyramid

Number of details

System

Multi-disciplinary

Mono-disciplinary

System requirements

Design decisions

Parts, connections, lines of code

Research focus
How to go from System Level to Detailed Designs?

source
reticule
lens
wafer

overlay: 45nm
CD control: 10nm
productivity: 100Wph

10 Mloc
How to cope with dynamic range?
multiple people
overlap
stretch

How do skilled architects work?
viewpoint hopping
fast iterations
from vague to tangible
integrating
Major Bottleneck: Mental Dynamic Range

Architectural Thinking
15 Gerrit Muller

version: 1.1
March 6, 2013
ATmentalDynamicRange
From Components to System Qualities

**Components**
- Source
- Mirrors
- Fiducials
- Lens Elements
- Flaps
- Air Showers
- Frames
- Motors
- Sensors
- Robot
- Bolts
- Nuts

**Subsystems**
- Laser
- Lens
- Stages
- Handlers
- Electronics Infra
- Metro Frame

**Functions**
- Prepare
- Transport Wafer
- Align and Calibrate
- Scan and Expose
- Transport Reticle

**System Qualities**
- Overlay
- CD Control
- Productivity

Architectural Thinking

Gerrit Muller
How does an Architect Work? Experience Based!

thinking path of an architect during a few minutes up to 1 day

Architectural Thinking
17  Gerrit Muller

version: 1.1
March 6, 2013
BWMAexplorationPath
**Examples of Experience Based Methods**

*methods successfully applied in multiple domains:*
- key driver model;
- context modeling;
- cost of ownership modeling;
- use cases, worst cases
- graph representation for logistics purposes (commercial, goods flow, service)
- mapping functions to products and others (QFD)
- interface specification
- construction decomposition
- functional decomposition
- designing with multiple decompositions
- execution architecture
- performance modeling
- micro benchmarking
- **budget-based design**
- safety, reliability and security analysis, for example FMEA
- work break down structure
- integration plan
- quality checklist
- story telling

*domains where these models have been applied:*
- wafersteppers
- health care
- electronics infrastructure projects
- document handling
- consumer electronics
- semiconductors

*the budget-based design method will be discussed as applied in wafersteppers, health care, and document handling*

*this list of methods based on:*

example of implicit experience: budget based design
Why, What, How, Where, ...?
capture in explicit method
Budgets Applied on Waferstepper Overlay

- Reticule matching: 15 nm
- Lens matching: 25 nm
- Single machine: 30 nm
- Stage overlay: 12 nm
- Stage grid accuracy: 5 nm
- Metrology stability: 5 nm
- Global alignment accuracy: 6 nm
- Off-axis position measurement accuracy: 4 nm
- Stage Al. position measurement accuracy: 4 nm
- System adjustment accuracy: 2 nm
- Blue align sensor repro: 3 nm
- Interferometer stability: 1 nm
- Frame stability: 2.5 nm
- Tracking error WS: 2 nm
- Tracking error X, Y: 2.5 nm
- Tracking error RS: 1 nm
- Tracking error phi: 75 nrad

- Process overlay: 80 nm
- Matched machine: 60 nm
- Process dependency sensor: 5 nm
- Stage overlay: 12 nm
- Position accuracy: 7 nm
- Alignment repro: 5 nm
- Frame stability: 2.5 nm
- Tracking error WS: 2 nm
- Tracking error X, Y: 2.5 nm
- Tracking error RS: 1 nm
- Tracking error phi: 75 nrad
## Budgets Applied on Medical Workstation Memory Use

<table>
<thead>
<tr>
<th></th>
<th>code</th>
<th>obj data</th>
<th>bulk data</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>shared code</td>
<td>11.0</td>
<td></td>
<td></td>
<td>11.0</td>
</tr>
<tr>
<td>User Interface process</td>
<td>0.3</td>
<td>3.0</td>
<td>12.0</td>
<td>15.3</td>
</tr>
<tr>
<td>database server</td>
<td>0.3</td>
<td>3.2</td>
<td>3.0</td>
<td>6.5</td>
</tr>
<tr>
<td>print server</td>
<td>0.3</td>
<td>1.2</td>
<td>9.0</td>
<td>10.5</td>
</tr>
<tr>
<td>optical storage server</td>
<td>0.3</td>
<td>2.0</td>
<td>1.0</td>
<td>3.3</td>
</tr>
<tr>
<td>communication server</td>
<td>0.3</td>
<td>2.0</td>
<td>4.0</td>
<td>6.3</td>
</tr>
<tr>
<td>UNIX commands</td>
<td>0.3</td>
<td>0.2</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>compute server</td>
<td>0.3</td>
<td>0.5</td>
<td>6.0</td>
<td>6.8</td>
</tr>
<tr>
<td>system monitor</td>
<td>0.3</td>
<td>0.5</td>
<td>0</td>
<td>0.8</td>
</tr>
<tr>
<td><strong>application SW total</strong></td>
<td><strong>13.4</strong></td>
<td><strong>12.6</strong></td>
<td><strong>35.0</strong></td>
<td><strong>61.0</strong></td>
</tr>
<tr>
<td>UNIX Solaris 2.x</td>
<td></td>
<td></td>
<td></td>
<td>10.0</td>
</tr>
<tr>
<td>file cache</td>
<td></td>
<td></td>
<td></td>
<td>3.0</td>
</tr>
<tr>
<td><strong>total</strong></td>
<td></td>
<td></td>
<td></td>
<td>74.0</td>
</tr>
</tbody>
</table>
Power Budget Visualization for Document Handler

Architectural Thinking
version: 1.1
March 6, 2013
MDMpowerProportions
So, What are the Open Questions?

• generic nature of methods
  need for customization
  need for highly skilled designers

• lack of description
  concepts
  how tos
  where to learn (graduate, postgraduate, postdoc)?
  which discipline?

• lack of education in this type of methods

• lack of research (exploration and consolidation)
  when to apply?
  what are the limits?
  what are alternative methods?
  what are the options for (partial) solutions?

• lack of relation with mono-disciplinary methods
  how to use the results, f.i. how to transform a construction decomposition into a class decomposition?

• lack of tools?
Method abstraction hierarchy

principle — drives → method — elaborated in → recipe — supported by → notation template

abstract → specific and executable → tool
Attributes of a Method

- a goal
- a decomposition in smaller steps
- possible orders of taking these steps
- visualization(s) or representation(s)
- guidelines

- to make the design explicit
- to provide a baseline to take decisions
- to specify the requirements for the detailed designs
- to have guidance during integration
- to provide a baseline for verification
- to manage the design margins explicitly
## Budget Based Design Decomposition and Order

<table>
<thead>
<tr>
<th>Step</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1A</td>
<td>measure old systems</td>
</tr>
<tr>
<td>1B</td>
<td>model the performance starting with old systems</td>
</tr>
<tr>
<td>1C</td>
<td>determine requirements for new system</td>
</tr>
<tr>
<td>2</td>
<td>make a design for the new system</td>
</tr>
<tr>
<td>3</td>
<td>make a budget for the new system:</td>
</tr>
<tr>
<td>4</td>
<td>measure prototypes and new system</td>
</tr>
<tr>
<td>5</td>
<td>Iterate steps 1B to 4</td>
</tr>
</tbody>
</table>
Visualization of Budget Based Design Flow

- Measurements:
  - Existing System
  - Existing System Components
  - Micro Benchmarks
  - Aggregated Functions
  - Applications

- Design Estimates:
  - Simulations
  - V4aa

- Specifications:
  - SRS
  - t_{boot} = 0.5s
  - t_{zap} = 0.2s

- Budget:
  - t_{proc}
  - t_{over}
  - t_{disp}
  - T_{proc} = 30
  - T_{disp} = 25
  - T_{total} = 55

- Model:
  - t_{proc}
  - t_{over}
  - t_{disp}

- Feedback:
  - Measurements
  - New (Proto) System

- Architectural Thinking
  - Architectural Thinking Components
  - Architectural Thinking System

- Can be more complex than additions

---

Architectural Thinking

version: 1.1
March 6, 2013
EAAbudget
The design of Embedded systems
software intensive
large development teams
multi-disciplinary

will always be

Systems Architecting \( \times \) an art
Skilled architects are scarce

will become even more complex :

> 1 Mloc
10..1000fte
software, electronics,
physics, mechanics,...

> 10 Mloc
(10..100 suppliers)*(10..1000fte)
multi-system, multi-application,
multi-site, multi-*

explicit know how facilitates more engineering,
supported by more system engineers

Failures, delays, non-performance,
cost overruns, dissatisfied customers

plenty of research opportunities!
Research questions

• What are potential applications for budgets?
• What kind of budget is required?
• What is the decomposition to be used?
• How to manage margins?
• How to verify a budget?
• How to use and maintain a budget?
• Does it provide value when a budget is coupled to other design information?
• and many more...
Potential Applications of Budget based design

• resource use  (CPU, memory, disk, bus, network)
• timing (response, latency, start up, shutdown)
• productivity (throughput, reliability)
• Image Quality parameters (contrast, SNR, deformation, overlay, DOF)
• cost, space, time
What kind of budget is required?

<table>
<thead>
<tr>
<th>static</th>
<th>dynamic</th>
</tr>
</thead>
<tbody>
<tr>
<td>typical case</td>
<td>worst case</td>
</tr>
<tr>
<td>global</td>
<td>detailed</td>
</tr>
<tr>
<td>approximate</td>
<td>accurate</td>
</tr>
</tbody>
</table>

is the budget based on wish, empirical data, extrapolation, educated guess, or expectation?