Multi-view Architecting

by Gerrit Muller

Philips Research IST - SWA - IA

Abstract

Many people expect from the system architect that he decomposes the system in smaller components and defines and guards the interfaces. The conventional waterfall model for software development and this architecture view form a dangerous combination: an extremely limited integral understanding with a very late feedback.

A multi-view architecting approach tackles the problem of integral understanding. In combination with spiral or incremental development models a powerful method becomes available for creating complex systems.
Illustration case: MRI scanner
Block diagram view

Host

Acquisition
- magnet
- gradients
- RF
- ADC

Reconstruction

Storage

Viewing & Printing
Physics view

Gradient Coil
40 mT/m/ms

Magnet
3T, ...

RF coil, Eff vol. =
Software architecture view

Property editor
NameSpace server
Event manager
Registry
Persistent Storage
Application
Session manager
Broker
Transparant Communication
Monitor
Plug-in framework
Compliance profile
Device independent format
Spool server
Queue manager
Resource scheduler
Configurable pipeline
Abstraction Layer
Plug & play

Multi-view Architecting
Gerrit Muller
version: 0
25th September 2001
MVmechanismArchitecture
MR imaging methods view
Conceptual Work by the architect

- Most disciplines require multiple views, for instance circa 4 views in SW [Kruchten, Soni]
- Only a subset of disciplines has been shown (not shown are a.o. mechanics, logistics, project management)

The system architect integrates the complementing disciplinary views

However

Decisions and trade-offs in the conceptual view are driven by application, business and operational inputs
Useability and main stakeholders

The engineer creates a technological UI...

without imagining the clinical reality

"In the meantime the patient is horrified by the intimidating system, the weird cage around his body and the EKG leads attached to his breast..."

Select Virtual Representation Display Mode

- Intermittent
- Adaptive
- Semi-Reflective

Fuzzle Factor

Patient Jansen has been removed

OK
Radiology department view

MR Examination room

Control room

"MPR" room

CT Examination room

Control room

Reading Room
System Architect integrates 5 viewpoints

What does Customer need in Product and Why?

Customer

What

Customer objectives

How

Application

Product

What

Functional

Product

How

Conceptual

Realization
Integration of 5 views

- Customer business
- Application
- Functional
- Conceptual
- Realization

Integrating Views

- High margin Cardiology market
- Cost per examination model
- Patient throughput features
- System throughput model
- Budget in seconds
From scenario to budget

<table>
<thead>
<tr>
<th>Customer business</th>
<th>Application</th>
<th>Functional</th>
<th>Conceptual</th>
<th>Realization</th>
</tr>
</thead>
</table>

Scenario:

Accessible story, clearly outlining a frequently occurring situation with a valuable, but challenging solution

Typical Case:

Functions and Quantification of frequently occurring important and critical case

Models

Functional and Performance

Technical estimates

Several iterations are required. In later iterations worst cases and exceptional cases are taken into account. The technical estimates are then transformed in budgets.
MR neuro scenario

- Patient George has continuous headache.
- His family doctor has send him to the Neurologist.
- The Neurologist wants to exclude the possibility of a tumor and requests an MRI examination.
- The Radiologists does not see any indication for a tumour.
- The Radiologist sends his report to the Neurologist.
- The Neurologist discusses his findings with the patient and sends a report to the family doctor.
Clinical Stakeholders

- Family
- Doctor
- Patient
- Referring Physician
- Nurse, operator
- Radiologist

Processes:
- Consult
- Findings
- Request
- Report

MRI scanner
Typical timing of Neuro examination

14:00
George arrives at radiology department

14:00 - 14:15
Examination of previous patient

14:15
Nurse explains the procedure

14:15 - 14:30
15 minute time slot

14:15
George is waiting in the dressing room

14:15 - 14:30
Prepare George for the examination (a.o. RF coils)

14:30
George leaves exam room

14:30
Exam room

Position

Imaging

View away

View away
Typical amount of Images: 2 Volumes

Data in bytes =

\[ 2 \times 512 \times 512 \times 256 \times 2 = \]

Volumes \( x \times y \times z \) bytes per pixel

256 MBytes

in 2 * 2 minutes =
240 seconds
MR resource model

Acquisition ▶ Recon-struction ▶ Viewing

Intermediate data: 256 MByte

View away in ca 10 sec.

Storage
2 Volumes 256 MByte

Full screen 25 images per second
MR critical design choices

Attribute access
- Acquisition
- Reconstruction

Buffer architecture
- Intermediate data: 256 MByte
- Storage
  - 2 Volumes 256 MByte

Pipeline & caching
- Viewing

View away in ca 10 sec.
- full screen 25 images per second
Checklists for integrating 5 views

Customer business

Application

Functional

Conceptual

Realization

keydrivers

entities operations

functions features

functional blocks SW components

Customer business drives Application, which in turn drives Functional, which is fulfilled by domain specific considerations. Functional is then implemented by Conceptual, which is refined by Realization.

Customer business context influences stakeholders, who deal with entities and operations, which are constrained by functional blocks. System qualities are influenced by design concerns, which are refined by technical structures and mechanisms.

The process is guided by select mechanisms, which are addressed by system qualities.

The checklists are generic enough to be supported by checklists.
## Actual checklists

### Customer business
- Who decides?  
- Who pays?

### Application
- Consumer User  
- Operator Retailer

### Functional
- Safety  
- Security  
- Reliability  
- Robustness  
- Manufacturability  
- Testability  
- Serviceability  
- Configurability  
- Installability  
- Evolvability  
- Portability  
- Upgradeability  
- Extendability  
- Maintainability  
- Useability  
- Appeal, Appearance  
- Throughput or Productivity

#### Operational Stakeholders:
- Sales person  
- Field service engineer  
- Marketeer  
- Portfolio manager  
- Project manager  
- Developer

#### Relations
- Response Time  
- Image Quality  
- Reproduceability  
- Predicatability  
- Accuracy  
- Cost price  
- Cost of operation  
- Interaction with environment  
- Power consumption  
- Consumption rate (water, air, chemicals, etcetera)  
- Disposability  
- Size, weight  
- Resource utilization  
- Logistics flexibility  
- Lead time  
- Standards Compliance

### Conceptual
- Granularity, Scoping, Containment, Cohesion, Coupling  
- Interfaces, Allocation, Bugets  
- Information model (entities, relations, operations)  
- Identification, Naming  
- Static characteristics, Dynamic behavior  
- System level infrastructure  
- Software development process, Environment, Repository, Tools  
- Feedback tools (for instance monitoring, statistics and analysis)  
- Persistence  
- Licensing, SW-keys  
- Set-up sequence  
- Technology choices  
- Make, Outsource, Buy, or Interoperate decisions  
- Meta-functional aspects:
  - Operational (e.g. image processing, handling calls, …)
  - Initialization, Start-up, Shutdown  
  - Fault handling  
  - Diagnostics  
  - Configuration handling  
  - Data replication  
  - Performance observation  
  - Capability query  
  - Testing  
  - Debugging  
  - Off-line guidance

### Realization
- Exceptions, Logs, Traces  
- Process, Tasks, Threads  
- Configuration management: Packages, Components, Files, Objects, Modules, Interfaces  
- Automated testing: Special methods, Harness, Suites  
- Signalling, Messaging, Call-back scheduling, Notification, Active data, Watchdogs, Time-outs  
- Locking, Semaphores, Transactions, Checkpoints, Deadlock detection, Roll-back  
- Identification, Naming, Data model, Registry, Scoping, Configuration database, Inheritance  
- Resource management, Allocation, Fragmentation prevention, Garbage collection  
- Persistence, Caching, Versioning, Prefetching, Lazy evaluation  
- Licensing, SW-keys, Bootstrap, Discovery, Negotiation  
- Call graphs, Message tracing, Object tracing  
- Distribution, Allocation, Transparency; Component, Client/Server, Multi-tier model

---

**Multi-view Architecting**  
**Gerrit Muller**  
**version: 0**  
**25th September 2001**  
**CAFCRplusChecklists**  
**PHILIPS Research**  
**IST - SWA - IA**
Coverage of MR neuro view

Customer business
Who decides?
Who pays?

Application
Consumer
User
Operator
Retailer

Functional
Granularity, Scoping, Containment, Cohesion, Coupling
Interfaces, Allocation, Budgets
Information model (entities, relations, operations)
Identification, Naming
Static characteristics, Dynamic behavior
System level infrastructure
Software development process, Environment, Repository, Tools
Feedback tools (for instance monitoring, statistics and analysis)
Persistence
Licensing, SW-keys
Set-up sequence
Technology choices
Make, Outsource, Buy, or Interoperate decisions

Conceptual
Response Time
Image Quality
Reproducability
Predicatability
Accuracy
Cost price
Cost of operation
Interaction with environment
Power consumption
Consumption rate (water, air, chemicals, etcetera)
Disposability
Size, weight
Resource utilization
Logistics flexibility
Lead time
Standards Compliance

Realization
Exceptions, Logs, Traces
Process, Tasks, Threads
Configuration management:
Packages, Components, Files, Objects, Modules, Interfaces
Automated testing: Special methods, Harness, Suites
Signalling, Messaging, Call-back scheduling, Notification, Active data, Watchdogs, Time-outs
Locking, Semaphores, Transactions, Checkpoints, Deadlock detection, Roll-back
Identification, Naming, Data model, Registry, Scoping, Configuration database, Inheritance
Resource management, Allocation, Fragmentation prevention, Garbage collection
Persistence, Caching, Versioning, Prefetching, Lazy evaluation
Licensing, SW-keys, Bootstrap, Discovery, Negotiation
Call graphs, Message tracing, Object tracing
Distribution, Allocation, Transparency; Component, Client/Server, Multi-tier model

Operational Stakeholders:
Sales person
Field service engineer
Marketeer
Portfolio manager
Project manager
Developer

Operational aspects:
Response Time
Image Quality
Reproducability
Predicatability
Accuracy
Cost price
Cost of operation
Interaction with environment
Power consumption
Consumption rate (water, air, chemicals, etcetera)
Disposability
Size, weight
Resource utilization
Logistics flexibility
Lead time
Standards Compliance

Meta-functional aspects:
Operational (e.g. image processing, handling calls,...)
Initialization, Start-up, Shutdown
Fault handling
Diagnostics
Configuration handling
Data replication
Performance observation
Capability query
Testing
Debugging
Off-line guidance

Multi-view Architecting
Gerrit Muller

IST - SWA - IA
Architects must increase customer side contribution

- Current Architects
- Required Architects

customer, business, application, functional, conceptual, realization