Multi-view Architecting; Illustrated by an MRI scanner

by Gerrit Muller       Buskerud University College

   e-mail: gaudisite@gmail.com
          www.gaudisite.nl

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
Multi-view Architecting; Illustrated by an MRI scanner

3        Gerrit Muller

version: 0.2
March 6, 2013
MRcomputerScienceView
Physics view

Magnet 3T, ...
Gradient Coil 40 mT/m/ms
RF coil, Eff vol. =
MR imaging methods view

![Diagram of MR imaging methods view]

- **RF**
- **Gz**
- **Gx**
- **Gy**

**TE**

**TR**

Typical TE: 5..50ms
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...

Select Virtual Representation Display Mode

- Intermittent
- Adaptive
- Semi-Reflective

0.2 Fuzzle Factor

Patient Jansen has been removed

OK

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..."
Radiology department view

MR Examination room

Control room

"MPR" room

CT Examination room

Control room

Reading Room

Multi-view Architecting; Illustrated by an MRI scanner

version: 0.2
March 6, 2013
Medical Imaging CT MR department

Gerrit Muller
System Architect integrates 5 viewpoints

What does Customer need in Product and Why?

- Customer objectives
- Application
- Functional
- Conceptual
- Realization

drives, justifies, needs

enables, supports
Integration of 5 views

- **Customer objectives**
- **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

**Customer objectives**

**Application**

**Functional**

**Conceptual**

**Realization**

**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.
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

MRI scanner

Multi-view Architecting; Illustrated by an MRI scanner
Gerrit Muller
version: 0.2
March 6, 2013
MRradiologyPatientAndPhysicians
Typical timing of Neuro examination

- George arrives at radiology department
- Nurse explains the procedure
- George is waiting in the dressing room
- Prepare George for the examination (a.o. RF coils)
- Position
- Imaging
- View away
- View away
- 15 minute time slot
- George leaves exam room
- Examination of previous patient

14:00
14:15
14:30
Typical amount of Images: 2 Volumes

Data in bytes =

\[2 \times 512 \times 512 \times 256 \times 2 = \text{Volumes} \times \text{x} \times \text{y} \times \text{z}\]  

\[256 \text{ MBytes}\]

in \(2 \times 2\) minutes = 240 seconds
MR resource model

Acquisition → Reconstruction

Intermediate data:
256 MByte

Storage
2 Volumes
256 MByte

Viewing

View away in ca 10 sec.
full screen
25 images per second
MR critical design choices

- Acquisition
- Reconstruction
- Intermediate data: 256 MByte
- Storage: 2 Volumes 256 MByte
- Viewing
- Pipeline & caching
- View away in ca 10 sec.
- Full screen 25 images per second
Checklists for integrating 5 views

Multi-view Architecting; Illustrated by an MRI scanner

19 Gerrit Muller

Key drivers influence context.

Entities deal with operations.

System quality concerns are addressed by design concerns.

Technical structures and mechanisms are refined by SW components.

Functional blocks are implemented by functional features.

Domain specific concerns are fulfilled by implemented SW components.

Generic enough to be supported by checklists.

Customer business drives stakeholders.

Application functional conceptual realization.

Customer context drives business stakeholders.

Design concerns select mechanisms.

Functional blocks refine mechanisms.

Technical structures and mechanisms guide design concerns.

System quality concerns qualify features.

Domain specific concerns are fulfilled by implemented SW components.
## 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

### 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

### Realisation
- 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; Illustrated by an MRI scanner
Gerrit Muller
version: 0.2
March 6, 2013
CAFCRplusChecklists

---
## Coverage of MR neuro view

<table>
<thead>
<tr>
<th>Customer business</th>
<th>Application</th>
<th>Functional</th>
<th>Conceptual</th>
<th>Realization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Who decides? Who pays?</td>
<td>Consumer</td>
<td>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 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</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Safety Security Reliability Robustness Manufacturability Testability Serviceability Configurability Installability Evolvability Portability Upgradeability Extendability Maintainability Useability Appeal, Appearance Throughput or Productivity</td>
<td>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</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operational Stakeholders: Sales person Field service engineer Marketeer Portfolio manager Project manager Developer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Architects must increase customer side contribution

- Current Architects
- Required Architects

- customer objectives
- application
- functional
- conceptual
- realisation