How Reference Architectures support the evolution of Product Families; the Darwin research project

by Gerrit Muller  Embedded Systems Institute

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

Abstract

TBD

Distribution

This article or presentation is written as part of the Gaudi project. The Gaudi 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.

This work has been carried out as part of the Darwin project under the responsibility of the Embedded Systems Institute. This project is partially supported by the Netherlands Ministry of Economic Affairs under the BSIK program.
## High Level Problem Statement

<table>
<thead>
<tr>
<th>Installed Base Business</th>
<th>costly</th>
<th>high effort</th>
<th>diversity and # of configurations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Life Cycle Management</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Development efficiency</td>
<td>costly</td>
<td>high effort</td>
<td>too late</td>
</tr>
<tr>
<td>Innovation rate</td>
<td>too low</td>
<td>too late</td>
<td></td>
</tr>
</tbody>
</table>

How Reference Architectures support the evolution of Product Families

2 Gerrit Muller

version: 0.4
March 6, 2013
DERAhighLevelProblems
The Innovation Challenge

Challenge:
how to apply change locally for exploration of potential value and feasibility?

Postulate 1:
for effective exploration the following properties must be maintained:
- patient throughput
- system responsiveness
- image quality
- safety
- reliability

Postulate 2:
a system architecture that supports this level of exploration also supports the next phases of innovation: scaling-up and engineering

Postulate 3:
a system architecture that supports this level of exploration also supports life cycle business over many generations

potential innovation: change

inherently complex system e.g. MR scanner
Evolvability Problem Statement

**exploration is difficult**

- too much time, effort, cost
- from idea to tryout

**reliable realization is difficult**

- too much and unpredictable development time, effort, cost
- from tryout to realization

**engineering is difficult**

- some new features late relative to competition
- too much material and labor cost

---

**innovation life cycle**

- tryout: exploration of innovative features
- scale up for clinical use
- scale up for volume sales

---

How Reference Architectures support the evolution of Product Families

Gerrit Muller

version: 0.4
March 6, 2013
### Evolvability Problem Analysis

#### problems
- exploration is difficult
  - too much time, effort, cost
  - from idea to tryout
- reliable realization is difficult
  - too much and unpredictable development
time, effort, cost
  - from tryout to realization
- engineering is difficult
  - some new features late relative to competition
too much material and labor cost

#### observed causes
- 25 years of historical growth
- lack of overview
- size and complexity of realization
- inherent complexity of system and context
- human and cultural factors
  - high level of expertise
  - conservatism
- large amount of detailed documentation

#### suspected more specific root causes
- coupling (dependencies) higher than needed
- ineffective structure (decomposition, interfaces)
- insufficient underpinning of decisions by value and cost
- unbalance in core/key/base
- diversity of configurations

---

How Reference Architectures support the evolution of Product Families

Gerrit Muller

version: 0.4
March 6, 2013
DERAproblemAnalysis

---

**Evolvability Problem Analysis**

<table>
<thead>
<tr>
<th>problems</th>
<th>observed causes</th>
<th>suspected more specific root causes</th>
</tr>
</thead>
<tbody>
<tr>
<td>exploration is difficult</td>
<td></td>
<td></td>
</tr>
<tr>
<td>too much time, effort, cost</td>
<td></td>
<td></td>
</tr>
<tr>
<td>from idea to tryout</td>
<td></td>
<td></td>
</tr>
<tr>
<td>reliable realization is difficult</td>
<td></td>
<td></td>
</tr>
<tr>
<td>too much and unpredictable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>development time, effort, cost</td>
<td></td>
<td></td>
</tr>
<tr>
<td>from tryout to realization</td>
<td></td>
<td></td>
</tr>
<tr>
<td>engineering is difficult</td>
<td></td>
<td></td>
</tr>
<tr>
<td>some new features</td>
<td></td>
<td></td>
</tr>
<tr>
<td>late relative to competition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>too much material and labor cost</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25 years of historical growth</td>
<td></td>
<td></td>
</tr>
<tr>
<td>lack of overview</td>
<td></td>
<td></td>
</tr>
<tr>
<td>size and complexity of realization</td>
<td></td>
<td></td>
</tr>
<tr>
<td>inherent complexity of system and context</td>
<td></td>
<td></td>
</tr>
<tr>
<td>human and cultural factors</td>
<td></td>
<td></td>
</tr>
<tr>
<td>high level of expertise</td>
<td></td>
<td></td>
</tr>
<tr>
<td>conservatism</td>
<td></td>
<td></td>
</tr>
<tr>
<td>large amount of detailed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>documentation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>coupling (dependencies)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>higher than needed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ineffective structure (decomposition, interfaces)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>insufficient underpinning of</td>
<td></td>
<td></td>
</tr>
<tr>
<td>decisions by value and cost</td>
<td></td>
<td></td>
</tr>
<tr>
<td>unbalance in core/key/base</td>
<td></td>
<td></td>
</tr>
<tr>
<td>diversity of configurations</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Darwin Project Goal

- specifically methods, techniques and patterns to improve the evolvability of product families within industrial constraints and while maintaining other qualities.

- based on modeling and Reference Architectures.

- faster to market, less effort, more predictable.

- very relevant for MR, also relevant for others (partially) validated.

- market response to anticipated and unexpected changes.

- diverse products, installed base diversity.

- patient throughput, system responsiveness, image quality, safety, reliability.

- scientifically sound, suitable for PhD.

- very relevant for MR, also relevant for others (partially) validated.

- also relevant for others.
Darwin Research Model: Industry as Laboratory

- **source of inspiration**: Philips MR scanner
- **application playground**: industry
- **challenging problems**
- **apply new engineering methods**
- **observe results**
- **improve**
- **hypothesis**

**How Reference Architectures support the evolution of Product Families**

Gerrit Muller

version: 0.4
March 6, 2013
DERAindustryAsLaboratory
Sources of Change

customer context
- humans
- other systems
- legislation
- reimbursement

technical architecture
- clinical applications
- workflow applications
- domain specific technology
- generic technology

business architecture
- competition
- organization
- business model
Sources of Change

customer context
- humans
- other systems
- legislation
- reimbursement

technical architecture
- clinical applications
- workflow applications
- RF coils
- gradient amplifier

business architecture
- competition
- organization
- business model
- domain specific technology
- generic technology
- Windows Vista
- PCI-X
- database

PACS
RIS
PMW
PII
USA

DERAsourcesOfChangeAnnotated

version: 0.4
March 6, 2013
Darwin Research Questions

How to transform into an evolvable product family architecture?

How to support decision making?
- business wise
- technological

How to create overview?
- by visualization
- by high-level modeling

How to mine the realization for implicit know how?

What are practical guidelines?
- for decomposition
- for interface definition

What are patterns that support evolvability?

related research areas

- value analysis, e.g. real option
- roadmapping
- reference architecture
  - physical models, functional models,
  - budgeting, figures-of-merit,
  - state-diagrams, time-lines
- repository meta-data analysis
- dynamic dependency analysis
- semantic analysis
- reference architecture
  - physical models, functional models,
  - qualities, behavior models
  - clustering, structure, set-based design
RA = Business Arch. + Technical Arch. + Customer Context

customer context

technical architecture

relations
guidance

business architecture

customer enterprise
users

requirements
black box view

design patterns
technology

business model
life cycle

How Reference Architectures support the evolution of Product Families

Gerrit Muller

version: 0.4
March 6, 2013
SAFRAarchitectures
1. Functional Decomposition

2. Construction Decomposition

3. Allocation

4. Infrastructure

5. Choice of integrating concepts

How Reference Architectures support the evolution of Product Families

How Reference Architectures support the evolution of Product Families

version: 0.4
March 6, 2013

LWAarchitectureHow

Technical Architecture
Decomposition and Interfaces

How Reference Architectures support the evolution of Product Families

13  Gerrit Muller

version: 0.4
March 6, 2013

BWMA decomposition
Interface much more than functions + parameters

**black box** *(interface) level:*
- protocols
- behavior
- characteristics

**white box** *(implementation) level:*
- protocols
- realizations
- limitations
- constraints
- opportunities
- behavior
- characteristics
Integration and Diversity

**MR image acquisition**
- bore systems
- open magnets
- 7T
- 3T
- 1.5T
- 1T
- 0.6T
- RF coils
- integrated
- dedicated

**image handling**
- gradients
- very fast
- fast
- economy
- prepare diagnosis
- treatment planning
- diagnosis
- research
- report
- education
- authorise
- demonstration
- clinical review

**information handling**
- administration
- billing
- scheduling
- logistics
- laboratory
- pharcaceutics

How Reference Architectures support the evolution of Product Families
Gerrit Muller
Distribution Scenario’s

A. "Thin Servers"
   - Clients
   - Network
   - Thin Servers

B. "All-in-one" Combi's
   - All-in-one Combi’s
   - Network
   - legend: acquisition, image handling, information handling, generic technology

C. "All-in-one" server
   - Thin Clients
   - Network
   - All-in-one Server (PACS or HIS)

D. "Modular"
   - Client
   - Network
   - Server
   - Server
Simplistic Architecture

How Reference Architectures support the evolution of Product Families

17 Gerrit Muller

version: 0.4
March 6, 2013

ARMR simplisiticArchitecture
Future Simplistic Architecture

How Reference Architectures support the evolution of Product Families

version: 0.4
March 6, 2013

ARMsimplisticArchitectureFuture
How Reference Architectures support the evolution of Product Families

Gerrit Muller
Example Long Term Vision

Long Term Vision: Reference Architecture + Sample implementation of Framework and Components

- Applications
- Services
- Framework
- Computing Infrastructure
- Domain Infrastructure

How Reference Architectures support the evolution of Product Families

version: 0.4
March 6, 2013
Conclusion: Refactoring the Architecture is a must