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.

June 5, 2018

status: planned

version: 0.4
# High Level Problem Statement

<table>
<thead>
<tr>
<th>Installed Base Business</th>
<th>costly</th>
<th>diversity and # of configurations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Life Cycle Management</td>
<td>high effort</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Development efficiency</th>
<th>costly</th>
<th>high effort</th>
<th>too late</th>
</tr>
</thead>
</table>

| Innovation rate         | too low | too late    |

See next slides
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
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
- **1**: scale up for clinical use
- **10**: scale up for volume sales
- **100**: volume

---

How Reference Architectures support the evolution of Product Families

Gerrit Muller

version: 0.4
June 5, 2018
DERAproblemStatement
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
- size and complexity
  - of organization
- 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
Darwin Project Goal

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

- scientifically sound suitable for PhD
- based on modeling and Reference Architectures
- faster to market less effort more predictable
- market response to anticipated and unexpected changes
- very relevant for MR also relevant for others (partially) validated
- diverse products installed base diversity
- people, process, project duration, and cost
- patient throughput system responsiveness image quality safety reliability

How Reference Architectures support the evolution of Product Families

version: 0.4
June 5, 2018
DERAProjectGoal
Darwin Research Model: Industry as Laboratory

source of inspiration

Philips MR scanner

application playground

industry

challenging problems

apply new engineering methods

research

hypothesis

improve

observe results

evaluate
Sources of Change

- Business architecture
- Technical architecture
- Customer context

**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
- business architecture
  - competition
  - organization
  - business model

RF coils
gradient amplifier
- domain specific technology
- generic technology

PMW
PII
PACS
RIS
USA
Windows Vista
PCI-X
database
Darwin Research Questions

How to transform into an evolvable product family architecture?

<table>
<thead>
<tr>
<th>How to support decision making?</th>
<th>business wise technological</th>
</tr>
</thead>
<tbody>
<tr>
<td>How to create overview?</td>
<td>by visualization by high-level modeling</td>
</tr>
<tr>
<td>How to mine the realization for implicit know how?</td>
<td></td>
</tr>
</tbody>
</table>

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
  - Requirements
  - Black box view
- Technical architecture
  - Design patterns
  - Technology
- Relations
  - User relations
- Guidance
  - Business model
  - Life cycle
- Business architecture
1. Functional Decomposition

2. Construction Decomposition

3. Allocation

4. Infrastructure

5. Choice of integrating concepts

How Reference Architectures support the evolution of Product Families

version: 0.4
June 5, 2018
LWAArchitectureHow
Decomposition and Interfaces

How Reference Architectures support the evolution of Product Families

version: 0.4
June 5, 2018
BWMAdecomposition
Interface much more than functions + parameters

**black box (interface) level:**
- protocols
- functions
- parameters
- formats

**white box (implementation) level:**
- protocols
- realizations
- limitations
- constraints
- opportunities
- behaviors
- characteristics

How Reference Architectures support the evolution of Product Families

version: 0.4
June 5, 2018
ECMAblackWhite
Integration and Diversity

### MR image acquisition

- Bore systems
- Open magnets
- RF coils
- Gradients
- Very fast
- Fast
- Economy
- Integrated
- Dedicated
- 1T
- 1.5T
- 3T
- 7T

### Image handling

- Prepare
- Diagnosis
- Treatment
- Planning
- Diagnosis
- Research
- Report
- Education
- Authorise
- Demonstration
- Clinical
- Review

### Information handling

- Administration
- Billing
- Scheduling
- Logistics
- Laboratory
- Pharmaceutics

---

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

version: 0.4

June 5, 2018

Gerrit Muller
Distribution Scenario’s

A  "Thin Servers"

B  "All-in-one" Combi's

C  "All-in-one" server

D  "Modular"

Network

Client

All-in-one Combi's

Thin Clients

All-in-one Server (PACS or HIS)

Client

Client

Thick Servers

Network

Server

Server

legend

acquisition

image handling

information handling

generic technology
Simplistic Architecture

How Reference Architectures support the evolution of Product Families

17 Gerrit Muller
Future Simplistic Architecture

How Reference Architectures support the evolution of Product Families

version: 0.4
June 5, 2018
ARMRsimplisticArchitectureFuture
How Reference Architectures support the evolution of Product Families

Available Code Assets

How Reference Architectures support the evolution of Product Families

version: 0.4
June 5, 2018

ARMRcodeAssets
Example Long Term Vision

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

Reference Architecture

Applications

Services

Computing Infrastructure

Domain Infrastructure

personalization
i.e. tunes, themes

Configuration
i.e. Internationalization

How Reference Architectures support the evolution of Product Families

version: 0.4
June 5, 2018
Conclusion: Refactoring the Architecture is a must