From Legacy to State-of-the-art; Architectural Refactoring

by Gerrit Muller       University of South-Eastern Norway-NISE
                    e-mail: gaudisite@gmail.com
                    www.gaudisite.nl

Abstract

The market of electronic appliances shows a fast increasing diversity. Manufacturers must be able to combine existing functions and new applications in a short time frame. A large amount of accumulated SW code (legacy) has to be reused in new ways.

The architecture(s) must be adapted to these new ways of working. Revolutionary adaptations have proven to be extremely risky. Opportunistic extension and integration decrease the quality of the code base, making it increasingly more difficult to continue. Architectural refactoring is a feedback based method to evolve an architecture.
Today’s Audio Video Consumer Products

From: COPA tutorial, Rob van Ommering
Trend: Convergence of separate worlds
Integration and Diversity

GSM phone

firewall

dvd

audio microset

set top box

headphone

pda

sailboat

cable modem

car

car navigation

speech

mp3

television

pen

garment

camera

Garment

computer

Games

flat display

Ambient Intelligence

living room

Communicator

Ambient Intelligence

living room

From Legacy to State-of-the-art; Architectural Refactoring

Gerrit Muller

version: 1.3

September 1, 2020

LWAIntegrationAndDiversity
Today’s Video Products

From Legacy to State-of-the-art; Architectural Refactoring
5  Gerrit Muller

version: 1.3
September 1, 2020
ARcurrentProducts
Evolution of Video Products

From Legacy to State-of-the-art; Architectural Refactoring

Gerrit Muller

version: 1.3
September 1, 2020
ARepProductEvolution
Distribution Scenario’s

A "Thin Servers"

B "All-in-one" Combi's

C "All-in-one" Home server

D "Modular"

From Legacy to State-of-the-art; Architectural Refactoring
Gerrit Muller
version: 1.3
September 1, 2020
ARproductScenarios
Product Packaging Options

4A "Thin Servers"

Digital Cable
Set top
DVR
DVD RW
Gateway

4C "All-in-one" Home server

Digital Cable
Set top
DVR
DVD RW
Gateway

4B "All-in-one" Digital TV

Digital Cable
Gateway

4D "Modular"

Digital Cable
Gateway

From Legacy to State-of-the-art; Architectural Refactoring
Gerrit Muller
version: 1.3
September 1, 2020
ARpackaging
Moore’s law

From COPA tutorial, Rob van Ommering
Problem: increasing SW size, decreasing reliability?

![Graph showing the typical amount of errors per product from 1990 to 2005. The x-axis represents the years (1990, 1995, 2000, 2005), and the y-axis represents the manyears per product. The graph shows a steady increase in the typical amount of errors per product over time.]
The Holy Grail: Reuse

![](image)

<table>
<thead>
<tr>
<th>Promise</th>
<th>Reality</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{REUSE}$</td>
<td>$\text{time}$</td>
</tr>
</tbody>
</table>

From Legacy to State-of-the-art; Architectural Refactoring

version: 1.3
September 1, 2020
ARtheHolyGrail
Simplistic Architecting: Digital TV

**Digital TV**

Digital Video Platform SW

<table>
<thead>
<tr>
<th>TV domain HW</th>
<th>Set Top Box domain HW</th>
<th>Computing HW</th>
</tr>
</thead>
</table>

**Set Top Box**

3rd party stack(s) | Set Top Box functions | MHP |

<table>
<thead>
<tr>
<th>TV domain HW</th>
<th>Computing HW</th>
</tr>
</thead>
</table>

**analog TV**

TV applications | TV computing Infra-structure |

TV domain platform

**Digital Video Platform SW**

<table>
<thead>
<tr>
<th>TV domain HW</th>
<th>Set Top Box domain HW</th>
<th>Computing HW</th>
</tr>
</thead>
</table>

Digital TV UI

<table>
<thead>
<tr>
<th>Digital TV UI</th>
</tr>
</thead>
</table>

3rd party stack(s) | Set Top Box functions | MHP |

<table>
<thead>
<tr>
<th>TV domain platform</th>
</tr>
</thead>
</table>

**Merge**

From Legacy to State-of-the-art; Architectural Refactoring

version: 1.3

September 1, 2020

ARdigitalTelevisionSimplisticArchitecture

Gerrit Muller
Available Code Assets

<table>
<thead>
<tr>
<th>Digital TV UI</th>
<th>Set Top Box Platform</th>
<th>MHP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TV applications</td>
<td>TV computing Infrastructure</td>
<td>3rd party stack(s)</td>
</tr>
<tr>
<td>TV domain platform</td>
<td>&gt;200 Myr</td>
<td>&gt;100 Myr</td>
</tr>
<tr>
<td>&gt;100 Myr</td>
<td>Set Top Box Platform</td>
<td>&gt;100 Myr</td>
</tr>
<tr>
<td>Digital Video Platform SW</td>
<td>&gt;100 Myr</td>
<td></td>
</tr>
<tr>
<td>TV domain HW</td>
<td>Set Top Box domain HW</td>
<td>Computing HW</td>
</tr>
<tr>
<td>&quot;Legacy&quot; code &gt; 500 Myr</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Merge problems

Architectural mismatch:
wrappers, translators, conflicting controls

additional code and complexity, no added value

Poor performance; additional resource usage

Problems Architecture Reuse non problem
Solution: Architectural Refactoring

Refactoring

within short term business goals

with limited but substantial refactoring goals

limited investment

based on long term architecture vision

clear product

clear value proposition

feedback on direction

limited investment based on long term architecture vision
Example of Refactoring Goals

+ Decrease Code Size

+ Decrease Resource Usage
  * power
  * memory
  * silicon area

+ Increase Performance
  * response time
  * throughput

+ Increase quality
  * decrease fault density
Architectural vs Code refactoring

Architectural Refactoring
Function, Structure, Rationale

Mechanisms, Technologies

Code Refactoring

From Legacy to State-of-the-art; Architectural Refactoring
17 Gerrit Muller

version: 1.3
September 1, 2020
ARcodeVersusArchitecturalRefactoring
Frequent feedback
stepsize: 3 months
elapsed time: 25 months
Feedback (2)

stepsize: elapsed time

From Legacy to State-of-the-art; Architectural Refactoring
20  Gerrit Muller
Small feedback cycles result in Faster Time to Market
Prerequisite 2

Awareness of dynamics
Myth: Platforms are Stable

Dynamic Market

How stable is a platform or an architecture?

Fast changing Technology

Architecture Components Platform
3rd generation components are mature, active maintenance needed.
Growth and change continues, some "old" components become obsolete.
Prerequisite 3

Long Term Vision
Example Long Term Vision

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

From Legacy to State-of-the-art; Architectural Refactoring

26    Gerrit Muller

version: 1.3
September 1, 2020
ARlongTermVisionExample
Don’t do

Opportunistic Legacy Integration

Proclaimed reuse
Conclusion: Refactoring the Architecture is a must

TV

Hybrid TV

Digital TV

"All-in-one" combi TV

From Legacy to State-of-the-art; Architectural Refactoring
Gerrit Muller

version: 1.3
September 1, 2020
ARrefactoring