# Architecture Reuse and Related Questions.

**White Paper Resulting from Architecture Forum Meeting**

**October 7, 8 2009 (Helsinki, Finland)**

**Edited by:**

**Dr. Gerrit Muller**, Buskerud University College, Embedded Systems Institute

**Mr. Eirik Hole**, Stevens Institute of Technology

Input was provided by the following participants in the Architecture Forum:

| Name | Organization |
|---|---|
| Frank Benschop | Philips Healthcare |
| Mans Bjuggren[1] | Micronic Laser Systems AB |
| Petri Haavisto | Nokia |
| Miikka Kemppinen | Nokia Siemens Networks |
| Bjørn Victor Larsen | Kongsberg |
| Hugo van Leeuwen | FEI Company |
| Sari Leppänen | Nokia |
| Erik Kreuwels | FEI Company |

| Name | Organization |
|---|---|
| Gerrit Muller | Buskerud University College/ESI |
| Sølve Raaen | Kongsberg Maritime |
| Kari Sahlman | Nokia Siemens Networks |
| Rolf Siegers | Raytheon |
| Tapio Tallgren | Nokia Siemens Networks |
| Dinesh Verma | Stevens Institute of Technology |
| Osmo Vikman | Nokia |
| | |

**Published February 2012**

## 1. Introduction

Reuse is and has been a hot topic in recent decades. We decided to focus on architecture reuse rather than the well trodden area of reuse of implementations, components or software. Since we realize that these subjects are related we try to emphasize architecture reuse, with side steps in the other reuse issues, applying the 80/20 rule here.

The idea behind frameworks such as TOGAF, DoDAF, MODAF, NAF, and AGATE is to facilitate architecture descriptions that can be implemented, analyzed, and also reused. For example in TOGAF reusable architectural artifacts, like business capabilities and technical components are stored in a repository. In enterprise architectures the use of architecture principles is encouraged, which is a high level form of architecture reuse. The proliferation of architecture frameworks and the popularity of Enterprise Architectures raise the expectations of architecture reuse.

The Architecting Forum members explored architecture reuse based on the following questions:

1. What does your company do in [architecture] reuse?
2. Where do you struggle?
3. What are your critical success factors?

We also had a more detailed list of questions for further exploration:

1. What to reuse (implementations, concepts, architectures, ideas, ...)
2. What is the benefit?
3. How to find re-usable artifacts?
4. How to integrate?
5. How to facilitate?
6. How to assess?
7. How to organize? Governance
8. What business model, what incentives?

9. What granularity, what scope?
10. How to support, how to evolve?

## 2. Architecture Reuse Experiences

Most participating companies actively strive for asset reuse. In practice this translates in reusing existing components or subsystems. Sometimes components or subsystems are reused "as is", but often the existing part evolves further to fit better in the new system(s). These reused assets implement an architecture and may or may not have an explicitly documented architecture. In many cases the architecture reuse is a consequence of reusing assets rather than a driving force.

***Principle 9.1 In practice, it is often asset reuse that implicitly causes architecture reuse.***

We used examples by Philips, Nokia, and Kongsberg Maritime to discuss *what* and *how*, and *why* and *why not* of architecture reuse.

### What and How of Architecture Reuse

Interfaces, communication infrastructure, and synchronization mechanisms are very typical elements of architectures that are reused when the focus has been on asset reuse. These elements appeared in all the presentations of the participants about their architecture reuse.

For example, Kongsberg Maritime uses a common base system for *Dynamic Positioning* of ships, *Ship Automation*, and *Process Control* on ships. The common base provides communication and synchronization, and also provides an interface approach to connect to a diverse set of ship actuators and sensors. The common base system is realized as pre-integrated hardware and software building blocks that can be used by multiple application projects.

The Magnetic Resonance Imaging (MRI) group of Philips HealthCare showed the evolution and use of the synchronization concepts in MRI scanners, the so-called Stretch principle [Mehlkopf 1983]. The Stretch principle can also be viewed as an architecture pattern.

***Principle 9.2 Architecture patterns promise to be a natural way to achieve architecture reuse.***

Nokia produces a wide variety of cell phones in multiple product families. All these products share the same overall architecture, with varying implementations of subsystems and components in both hardware and software. For example, most products are configured by selecting the appropriate implementation for the core chip set (CPU, DSP, modem), memory, imaging and video capture, display, connectivity (Bluetooth, WiFi), (Symbian) OS, Middleware, applications, modem stack, and HW drivers and adaptation layer. Interface management is critical in configuring different products from available and evolving components. Many products share the exact same interfaces and versions. Most products maintain the same logical interfaces, even across their later versions.

The discussion of the cases triggered the observation that different parts of the system change at different time-scales. E.g. software and digital electronics technologies change rapidly (and hence the hardware/software architecture might evolve fast). Algorithms and concepts tend to change slower, and the human and physics context changes even more slowly.

Another observation was that systems over time collect an ever increasing legacy. At the one hand this means a high degree of (implementation) reuse, at the other hand the question is whether the architecture stays healthy with such amount of legacy.

**Why Architecture Reuse**

The rationale for architecture reuse follows the typical rationale for asset reuse: e.g. improved quality based on proven performance, improved time to market, reduced development cost or protection of past investments, reduced operational costs, risk reduction, branding, and competence sharing. Facilitating an ecosystem is more specific for architecture reuse, although this argument was mentioned as a positive (facilitating) and negative (constraining).

Potential disadvantages also follow the disadvantages of asset reuse:

- Potentially limiting innovations (e.g. moving from portable phone to iPhone)
- Lock-in of legacy realizations and concepts
- Might be inappropriate in changing business environment, e.g. changes in competition, vendors, features, customer expectations or needs, cost or business models
- Potentially limiting technology changes
- All benefits mentioned before might turn out to be the opposite, e.g. where we strive for cost reduction the architecture sharing might induce a cost increase

### 3. The human dimension

During the discussion, the role of the developers was emphasized. It was remarked that "Reuse mostly happens through humans". Humans carry knowledge and have the skills to apply the knowledge. Architecture reuse is an attempt to capture this knowledge in repositories such that the knowledge can be reused elsewhere. There is a clear tension between the idea to capture knowledge in documents or artifacts in repositories and the observation that humans carry knowledge and have the reuse capability.

Also related to the human dimension is the culture of an organization's workforce. It was observed that the forum participants already are strong opponents pro or contra certain approaches. One of the remarks during the meeting was: "Opponents to architecture reuse have many reasons to justify not doing it". Probably the reverse proposition is also valid: "Proponents to architecture reuse have many reasons to justify doing it". The challenge for us is to peel one more layer of the onion to find more objective insights in architecture reuse; we need to be able to quantify the benefits, e.g. financial, time-to-market, quality improvement.

## 4. Quality descriptions as re-usable artifacts?

The approach to qualities, such as reliability, safety, and performance, may serve as a starting point for future architectures. The relevance of qualities is that these system level properties are the result of the integration of all contributing parts. For a more complete list of qualities see < http://www.gaudisite.nl/QualityNeedlesSlides.pdf>.

Several companies have good experiences with brief write-ups (2 to 5 pages) of each program's approach to addressing qualities. In many cases, the external properties are realized by applying internal design aspects. For example, performance as quality is achieved by resource management, and security is supported by data protection, and layering and partitioning.

One of the commonly shared challenges is the configurability and customizability during the life cycle, see Figure 1. The life cycle starts with development. The deliverables from the development are used by different projects, creating product variations. When deployed in the field the systems are further configured by multiple parties: sales or service, 3rd parties, and customers. Systems also do run-time configurations, such as calibrations. Complicating factors are changes in the field, for example triggered by Engineering Change Orders (ECO). The architecture is the means to manage compatibility, for example by versioning.

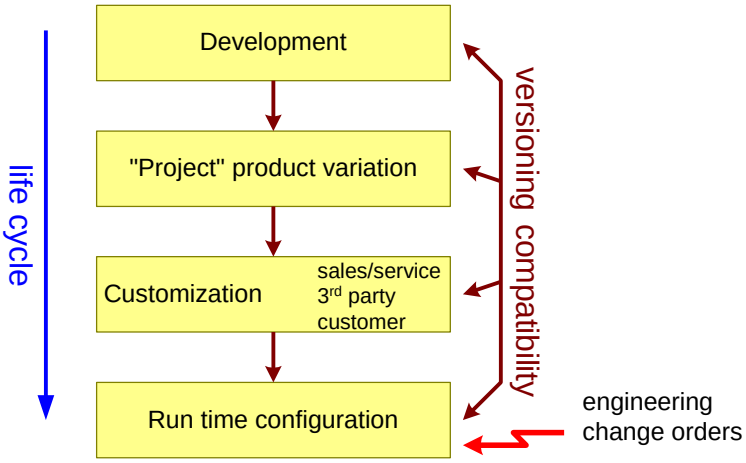


Figure 1, configurability and customizability during the life cycle

### 5. Other potential architecture reuse artifacts

We made a brief inventory of artifacts that could facilitate architecture reuse. We asked for three categories: what have you seen, what are you trying, and what do you plan. The answers were:

**What have you seen?** Copy/paste of complete descriptions or designs, training modules, tooling, data formats, stakeholder requirements, physical structures, and interface definitions.

**What are you trying?** Architecture rules, HMI guidelines, test automation, and checklists.

**What do you plan?** SysML models, unique error codes

### 6. Architecture reuse via artifacts or documents

Companies that actively promote architecture reuse emphasize documents, artifacts, and repositories. Documents and artifacts capture knowledge that is stored in repositories and/or databases. Storing knowledge in this way only makes sense when users access and use this data. The tools are often complemented by human measures focusing on:

- Experience level of practitioners
- Training on techniques, standards, and tools
- Awareness of available shared assets
- Collaboration internally and externally, promoting experience and idea exchange

We can store and share assets and asset related artifacts. However, we can also capture experiences as lessons learned in databases. Any database or repository needs good search capabilities to assist users in finding the data they are looking for. This may sounds as an open door but, unfortunately, search capabilities of many in-house repositories tend to be much less effective than Google searches on the public Internet.

Another common struggle is the tooling itself. The market of repository tools is still young, so some companies use custom-made tools. Such custom made-tool can be fully tailored to local needs. However, custom-made has the penalty that the tools need to be maintained. These tools often use proprietary formats and interfaces. Hybrid solutions, for example a third party

tool that is highly customized, are also common. The degree of customization can be so high that this solution is close to a custom-made tool. Several times architects complained about the complexity of tooling and advocated the KISS principle for repository tooling.

Repositories need to be embedded in the organization, which is quite a challenge. Taxonomy of the repository and metadata must fit the organization and its processes. Governance, ownership, Intellectual Property rights, and security levels need to be settled. The source and fidelity of data needs to be clear, and configuration management must be defined and done.

Success factors for repositories are ease-of-use (e.g. search capabilities), the balance between cost for inclusion in the repository and the benefit of reuse, and other incentives to get assets included or reused. Emotional factors play a role too; a "Not Invented Here" culture may hamper successful deployment.

## 7. Product Line and Reference Architectures

A complementary approach to architecture reuse is to "lift" architecture to a higher level, such as product line or reference architecture. See the SAF whitepaper of March 2007 [Hole 2007] for a discussion of Reference Architectures.

Architecture Standards can also serve as architecture reuse mechanism, where the standards body acts as architect. System architects can influence the content and evolution of the standards leveraged by their organization through participation in standards bodies.

## 8. Summary and Conclusions

We summarize by revisiting the questions posed at the beginning.

1. What does your company do in [architecture] reuse?

Reuse is clearly a hot topic. Most focus is on asset reuse, especially implementation assets. The asset reuse often implicitly causes architecture reuse. Architecture reuse via reference architectures or patterns is seen as promising, but the area is too young to have any proof of it validity. Tools such as asset repositories or lessons learned databases are operational, but several challenges from technical (searching capabilities) to organizational (governance, ownership, incentives) need still to be solved.

2. Where do you struggle?

Most companies feel pressure for their products' time-to-market and their overall financial results. Architecture reuse is a strategic approach; how do you convince decision makers to invest in this long-term thinking for future benefit?

3. What are your critical success factors?

Too little experience is present to actually make a well-founded list of success factors at this moment.

The general feeling is that architecture reuse is a promising way to harvest past experience and in that way speed up and increase the quality of future developments. However, at the same time we observe that operational pressure dominates and interferes with attempts to initiate architecture reuse activities. Our current experience level is insufficient to answer the more elaborated questions that we formulated at the beginning.

## Acronyms and Abbreviations

| | |
|---|---|
| AGATE | French framework **A**telier de **G**estion de l'**A**rchi**TE**cture des systèmes d'information et de communication (from http://en.wikipedia.org/wiki/AGATE_(architecture_framework)) |
| DoDAF | Department of Defense Framework (USA) |
| MODAF | Ministry of Defence Framework (British) |
| NAF | NATO Architecture Framework |
| TOGAF | The Open Group Architecture Framework |

## Literature

[Hole 2007] Reference Architectures; Why, What and How White Paper Resulting from Architecture Forum Meeting March 12-13, 2007 (Hoboken NJ, USA) Edited by: Gerrit Muller, Embedded Systems Institute / Eirik Hole, Stevens Institute of Technology. http://www.architectingforum.org/whitepapers/SAF_WhitePaper_2007_4.pdf

[Mehlkopf 1983] A.F. Mehlkopf, A. Bax, J. Smidt and T.A. Tiggelman, *Requirements, Design, and Specifications of a Multipurpose 7 Tesla NMR-system*, Computer Enhanced Spectroscopy, 1, 131, 1983