

Software Architecture Knowledge Management

Hans van Vliet
VU University, Amsterdam, The Netherlands
email: hans@cs.vu.nl



© The GRIFFIN Project

ASWEC 2008, 28-3-2008

1



© The GRIFFIN Project

ASWEC 2008, 28-3-2008

2

Software architecture, early definition

The architecture of a software system defines that system in terms of computational components and interactions among those components.

(from Shaw and Garlan, *Software Architecture, Perspectives on an Emerging Discipline*, Prentice-Hall, 1996.)



© The GRIFFIN Project

ASWEC 2008, 28-3-2008

3

Software Architecture

statement
↓
procedure
↓
module
↓
(design) pattern
↓
architecture

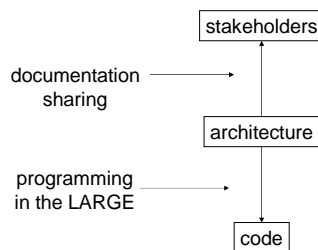


© The GRIFFIN Project

ASWEC 2008, 28-3-2008

4

Two flavors



© The GRIFFIN Project

ASWEC 2008, 28-3-2008

5

View models

- Kruchten's (now RUP) 4+1 model
- Siemens' 4 views model
- Philips BAPO (5 views)
- Zachman (36 views)
- ...




© The GRIFFIN Project

ASWEC 2008, 28-3-2008

6

Why Is Architecture Important?

- Architecture is the vehicle for stakeholder communication
- Architecture manifests the earliest set of design decisions
 - Constraints on implementation
 - Dictates organizational structure
 - Inhibits or enable quality attributes
- Architecture is a transferable abstraction of a system
 - Product lines share a common architecture
 - Allows for template-based development
 - Basis for training



(Bass et al, Software Architecture in Practice, 2003)

© The GRIFFIN Project ASWEC 2008, 28-3-2008 7

Where did it start?

- 1992: Perry & Wolf
- 1987: J.A. Zachman; 1989: M. Shaw
- 1978/79: David Parnas, program families
- 1972 (1969): Edsger Dijkstra, program families
- 1969: I.P. Sharp @ NATO Software Engineering conference:

“I think we have something in addition to software engineering [...] This is the subject of software architecture. Architecture is different from engineering.”

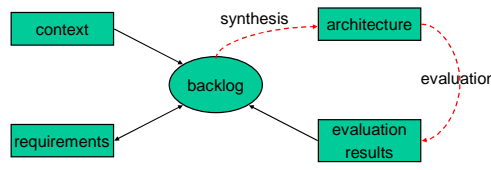
© The GRIFFIN Project ASWEC 2008, 28-3-2008 8

Design, design decisions

- Design is a ‘wicked problem’
 - Solutions are not true or false; there are only acceptable solutions
 - There’s no stopping rule
 - Every wicked problem is a symptom of another problem
- Design decisions:
 - There’s no linear sequence of design decisions
 - There’s often a series of alternatives to choose from
 - Design decisions incur further issues, and decisions

© The GRIFFIN Project ASWEC 2008, 28-3-2008 9

Global workflow in architecture design



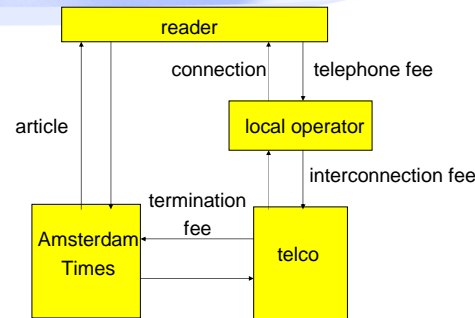
© The GRIFFIN Project ASWEC 2008, 28-3-2008 10

Design decision example: modeling a design space

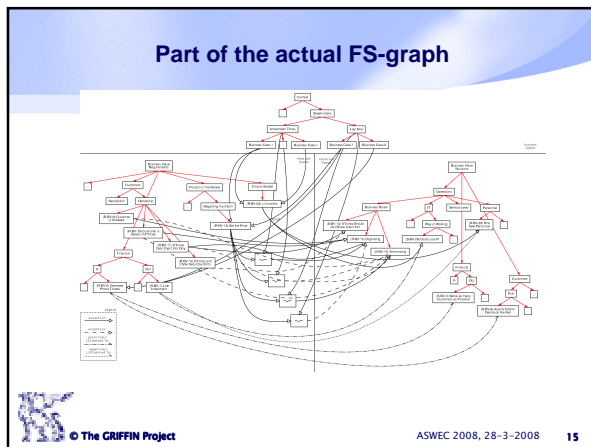
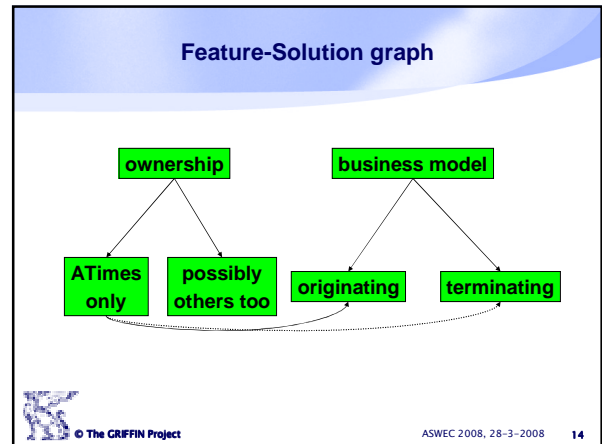
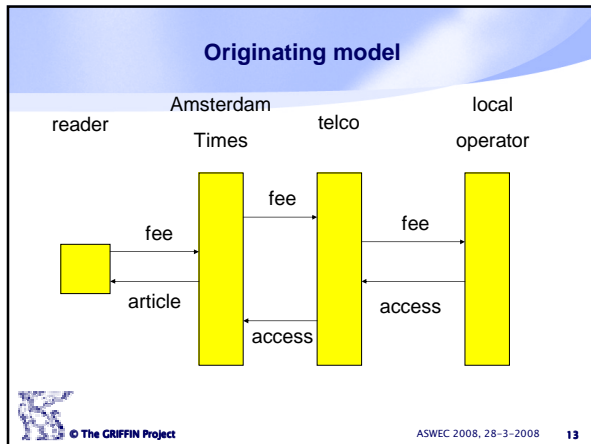
- Value-added news service: *Amsterdam Times* offers subscribers ability to read articles online
- Expenses to be covered by telephone connection revenues
- Critical issue: how value (money, information) flows between actors
- Two e-business models: terminating model (reader pays intermediate local operator) and originating model (reader pays *Amsterdam Times* directly)

© The GRIFFIN Project ASWEC 2008, 28-3-2008 11

Terminating model

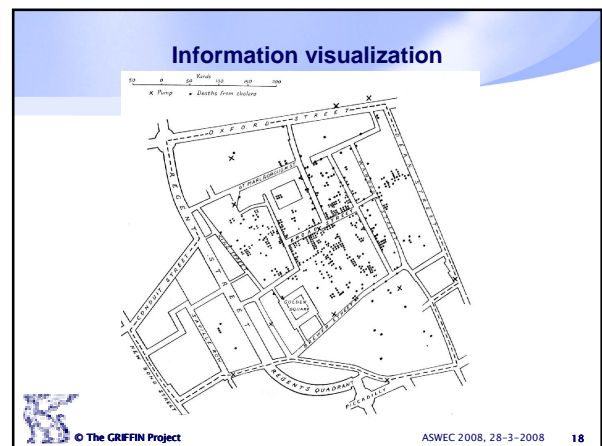


© The GRIFFIN Project ASWEC 2008, 28-3-2008 12



- ### Lessons learned
- Establishing business cases (“color FS graph”) did not result in new business models
 - The architect had “forgotten” about the reasoning steps and trade-offs involved in *getting* to the resulting models (knowledge vaporization)
- © The GRIFFIN Project ASWEC 2008, 28-3-2008 16

- ### Some questions to be raised
- Visualization?
 - How to induce architects to model a design space?
 - Architects do not search for optimal solutions, but for feasible ones
 - When does it help to model a design space?
 - Reasoning about the result?
- © The GRIFFIN Project ASWEC 2008, 28-3-2008 17



Hidden decisions

© The GRIFFIN Project ASWEC 2008, 28-3-2008 19

Overview

- Software Architecture
- Design Decisions
 - Case: modeling a design space
- **Architectural Knowledge: Griffin Project**

“What architects know and how they know it”

 - Design decisions versus requirements
 - Software architecture knowledge management
 - Software architecture and multi-site development
- Conclusions

© The GRIFFIN Project ASWEC 2008, 28-3-2008 20

AK crops up in different communities

- **Patterns:** patterns disseminate modeling and architectural knowledge among developers
- **Requirements:** AK plays a role in discussing the reciprocal relation between problem domain and solution space
- **Model-oriented:** AK is captured in a model that allows for formal reasoning
- **Software architecture:** AK is knowledge *about* a solution:
 - AK = solution (design) + design decisions + rationale

© The GRIFFIN Project ASWEC 2008, 28-3-2008 21

The Griffin Consortium

© The GRIFFIN Project ASWEC 2008, 28-3-2008 22

Research themes identified

Sharing

Improve the way AK is shared within and between organizations

Compliance

Support alignment to common architectural rules

Discovery

Enhance the findability of relevant Architectural Knowledge

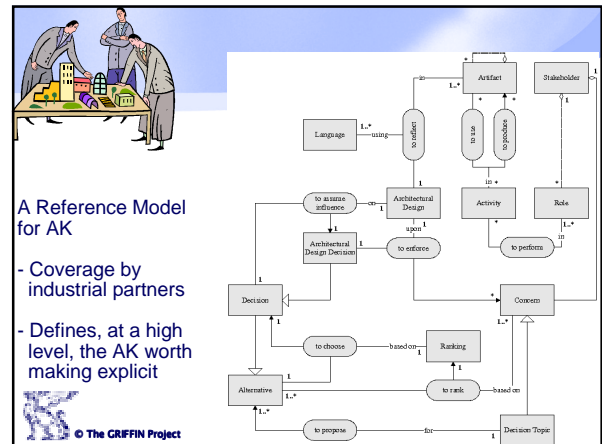
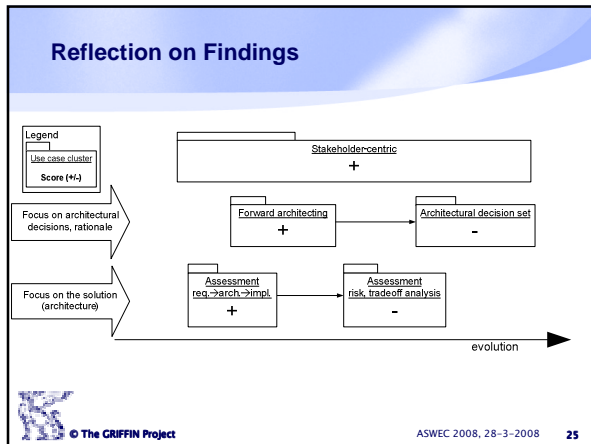
Traceability

Enable effective navigation through an organization's body of AK

© The GRIFFIN Project ASWEC 2008, 28-3-2008 23

A Use Case Model for typical AK usage

© The GRIFFIN Project



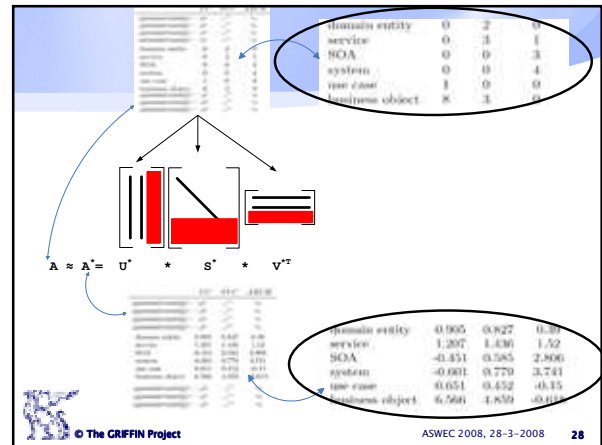
A Reference Model for AK

- Coverage by industrial partners
- Defines, at a high level, the AK worth making explicit

A Reading Guide for Newcomers

- **Problem:**
 - No reading guide for software product documentation
 - Where can one find relevant architectural knowledge?
- **Solution:**
 - Architectural Knowledge Discovery
 - 'Reading guide' for software product documentation
- **Technique: Latent Semantic Analysis (LSA)**
 - Uncovers latent semantic structure of documentation
 - Employ this structure to guide the reader

© The GRIFFIN Project ASWEC 2008, 28-3-2008 27



A Reading Guide: Where to start reading?

- Auditor knows nothing yet → interested in 'architecture' in a broad sense
- Preference of short documents over long documents
- **Q: What is the document that best matches the term 'architecture' ordered by document size?**
- **A: Document #46:**
 - 2 pages long,
 - does not contain the literal word 'architecture',
 - does provide an architectural overview (components, environment, relations, ...)

© The GRIFFIN Project ASWEC 2008, 28-3-2008 29

Importance of sharing architectural knowledge

"The major problem with intellectual capital is that it has legs and walks home every day." [Rus & Lindvall 2002]

- **Architecting is a very knowledge-intensive process**
 - Balance organizational, technical requirements and constraints
 - Communicate and motivate design decisions
 - Reuse architectural best practices whenever possible
- **AK sharing is important for various reasons**
 - Prevent loss of crucial architectural knowledge
 - Exchange experiences and ideas with colleagues
 - (Re)use architectural expertise and best practices
 - Train junior employees
- **Several tools for AK sharing suggested in literature,**
 - ... but few stories about their successful adoption in practice

© The GRIFFIN Project ASWEC 2008, 28-3-2008 30

Diagnosis: Desired approach to AK Sharing

- Integration**
 - Attractive and customizable GUI, link to internal and external information sources, notifications of new content (push)
 - Views
 - Content management, proper search
 - Discussion mechanisms

Need for Just-In-Time Architectural Knowledge:

"Access to and delivery of the right architectural knowledge, to the right person, at any given point in time."

- Community building**
 - Discussion mechanisms, overview of 'who knows what', share news, organize events.
- Constructing architecture descriptions**
 - Store and reuse best practices, intelligent decision-making support

© The GRIFFIN Project ASWEC 2008, 28-3-2008 31

Solution: EAGLE, a portal for AK sharing

Web-based, modular (open source) portal

- Main modules**
 - Document repository
 - Best practice database
 - Discussion boards
 - Yellow pages
 - Project environment
- Additional features**
 - More attractive and intuitive GUI
 - Integrated functionality
 - Stakeholder-specific content
 - Notifications and subscriptions

Supports hybrid AK sharing (codification + personalization)

Enables 'Just-in-Time' Architectural Knowledge

© The GRIFFIN Project WICSA 2008 / Vancouver 2008 32

Emerging issues

- Relation architectural knowledge/decisions ↔ requirements
- Role of architectural knowledge in GSD
- Codification ↔ personalization
- AK as metadata to documents; enrichment

© The GRIFFIN Project ASWEC 2008, 28-3-2008 33

Architectural decisions vs. requirements

- Architectural Decisions**
 - The logical view is organized in 3 layers: Data layer, Business logic layer, User-Interface layer.
 - Communication between classes uses RMI (Remote Method Invocation).
 - The system does not use MySQL as its relational database system.
 - The system is developed in Java.

(Philippe Kruchten)
- Architectural Requirements**
 - the system must be developed using Ada
 - the system must ensure that personal information is never made available without authorisation
 - Which type of DBMS is to be used? (Kotonya & Sommerville)

(Van Vliet)

© The GRIFFIN Project ASWEC 2008, 28-3-2008 34

Architectural knowledge and GS

- Architecture as proxy for communication/coordination
- More sites, larger system ⇒ more architecture
- Many RE GSD practices are applicable to architecture as well:
 - Frequent interaction across sites
 - Cross-site delegation ("traveling architects")
 - Collocated high-level architecture team
 - ...
- Emphasis on personalization aspects

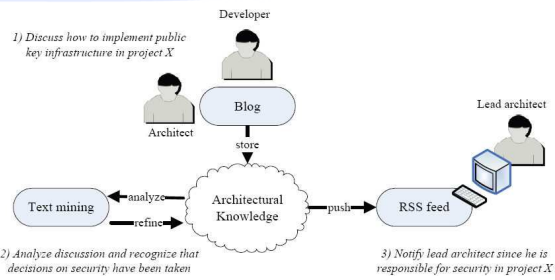
© The GRIFFIN Project ASWEC 2008, 28-3-2008 35

Codification and personalization

- You need both
- Current tools emphasize codification (ICT trap of KM)
- EAGLE prototype:**
 - Best practices repository
 - Document repository
 - Yellow pages
 - Discussion boards
 - Project environments
 - Blogs and wikis
- Vision: blackboard-like system with a.o. text mining, data enrichment services**

© The GRIFFIN Project ASWEC 2008, 28-3-2008 36

Automated refinement and enrichment of AK



© The GRIFFIN Project

WICSA 2008 / Vancouver 2008 / Bay 2008 37

Conclusions

- Software architecture = set of design decisions
- Shift to knowledge management perspective
- How to make SA knowledge management *effective*?



© The GRIFFIN Project

ASWEC 2008, 28-3-2008 38

Credits

(in alphabetical order)

- Remco de Boer (Discovery)
- Viktor Clerc (Compliance)
- Rik Farenhorst (Sharing)
- Anton Jansen (Traceability)

Scientific guidance

- Paris Avgeriou
- Patricia Lago
- Hans van Vliet (Project coordinator)



© The GRIFFIN Project

<http://griffin.cs.vu.nl>

ASWEC 2008, 28-3-2008 39