The Changing Face of Software Architecture ...

... and what that means for educators

David Garlan
12 March 2010

A Brief Personal History

- 1987 PhD at CMU
- 3 years at Tektronix developing a product line
- 1990 joined faculty at CMU
  - Began collaboration with Mary Shaw
  - Became involved in the Master in Software Engineering program
- 1992 taught first course in Architectures for Software Systems with Mary Shaw
- 1996 published book with Mary Shaw

12 March 2010
Acknowledgements

... Mary Shaw
... Tony Lattanze
... other co-instructors of Software Architecture course
... members of the ABLE research group
... funders (NSF, DARPA, ARO, Siemens, and many others)

12 March 2010

This Talk

- Software Architecture: past and present
  - What is software architecture?
  - Evolution of the field and its role in software engineering
- What should software engineers know about software architecture?
  - Elements of a course on software architecture
  - Architectural thinking
- Emerging trends and Issues
  - Architecture evolution
  - Architecture conformance
  - Frameworks, platforms, and ecologies
  - Conway’s law revisited
- Some questions to ponder

12 March 2010
Examples of Software Architecture

Figure 5.1: The Configuration of the GENESIS Prototype


12 March 2010
Figure 2. Display PostScript interpreter components.


12 March 2010
The Challenge

- Turn Software Architecture into an *engineering discipline*
  - from ad hoc definition to codified principles
- Develop systems “architecturally”
  - build systems compositionally from parts
  - assure that the system conforms to the architecture and has the desired properties
  - use standard integration architectures
  - reuse codified architectural design expertise
  - reduce costs through product-lines

12 March 2010

The Big Problem

How to bridge the gap between requirements and solutions?

12 March 2010
One Possible Answer

Requirements

A Miracle Happens!

- Ad hoc
- Requires gurus
- Unpredictable
- Costly

Implementations

The Role of Software Architecture

Requirements

Software Architecture

- High level of system design
- System-level abstractions
- Reuse design idioms

Implementations
What is Software Architecture?

- There are many definitions in the literature
  - CMU’s Software Engineering Institute’s web site on software architecture lists over 80 of them.
- The definition we like is this:

  The software architecture of a computing system is the set of structures needed to reason about the system, which comprise software elements, relations among them and properties of both.

Issues Addressed by Software Architecture - 1

- Gross decomposition of a system into parts
  - often using rich abstractions for component interaction (or system “glue”)
  - often using common design patterns/styles
- Emergent system properties
  - performance, throughput, latencies
  - reliability, security, fault tolerance, evolvability
- Rationale
  - justifying architectural decisions
- Envelope of allowed change
  - “load-bearing walls”
Software Architecture in Context

Evolution of the Field of Software Architecture – 1980’s

- Informal use of *box and line diagrams*
- Ad hoc application of architectural expertise
- Diverse, uncodified use of architectural patterns and styles
- No identified “architect” on most projects
1990’s

- Recognition of the value of *architects* in software development organizations
- *Processes* requiring architectural design reviews & explicit architectural documentation
- Use of *product lines*, commercial architectural *standards*, component *integration frameworks*
- *Codification* of vocabulary, notations & tools for architectural design
- *Books/courses* on software architecture

2000’s

- Incorporation of architectural notions into mainstream *design languages* and *tools* (e.g., UML-2)
- *Methods* based on architectural design and refinement (e.g., Model-Driven Design)
- Some architecture *analysis tools*
- Architectural *standards* for Enterprise Systems (e.g., RM-ODP, TOGAF)
- Architectural *frameworks* (e.g., SOA)
This Talk

- Software Architecture: past and present
  - What is software architecture?
  - Evolution of the field and its role in software engineering
- What should software engineers know about software architecture?
  - Elements of a course on software architecture
  - Architectural thinking
- Emerging trends and Issues
  - Architecture evolution
  - Architecture conformance
  - Frameworks, platforms, and ecologies
  - Conway’s law revisited
- Some questions to ponder

12 March 2010

What should software engineers know? -1

- General Concepts
  - What is software architecture
  - Basic concepts: views, styles, patterns
- Principles of Architecting
  - Understanding architectural requirements
  - Architecture styles and tactics
  - Product lines and integration frameworks
  - From architecture to code

12 March 2010
What should software engineers know? -2

- Architecture in Practice
  - Evaluating architectural designs
  - Handling architectural problems
  - Documenting a software architecture
  - Presenting an architecture to others
  - Architecture for X (security, usability, reliability, etc.)

Architectural Thinking - 1

An engineering mindset

Source: "The Google File System"
Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung

Figure 1: GFS Architecture
Different issues for architecture & programs

**Architecture**
- interactions among parts
- structural properties
- declarative
- mostly static
- system-level performance
- outside module boundary

**Programs**
- implementations of parts
- computational properties
- operational
- mostly dynamic
- algorithmic performance
- inside module boundary

Product lines, platforms, and styles

**Generic Styles**
- Data Flow
- Call-Return

**Generic Style Specializations**
- Pipes & Filters
- Process Control

**Generic Component Integration Standards**
- CORBA
- COM
- JavaBeans

**Domain-Specific Component Integration Standards**
- EJB
- HLA

**Product Line Frameworks**
- Tektronix Oscilloscopes
- Xerox Network Scanning
- Arch

Domain-Specificity
Knowing how much architecture is enough


Architectural Thinking - 5

Old styles never die ...
they just adapt to new technologies

Consider data-flow architectures
- In a Data Flow system
  - the design is dominated by orderly motion of data from process to process
  - the pattern of data flow is explicit
- Structural elements: data transformers & data channels
- Processing model: availability of data controls the computation
In the early days: Batch Sequential

Later: Pipes and Filters
Example: Apache

Source: Apache Modeling Project: Bernhard Gröne, Andreas Knöpfel, Rudolf Kugel, Oliver Schmidt

12 March 2010

Today: Yahoo! Pipes

12 March 2010
This Talk

- Software Architecture: past and present
  - What is software architecture?
  - Evolution of the field and its role in software engineering
- What should software engineers know about software architecture?
  - Elements of a course on software architecture
  - Architectural thinking
- Emerging trends and Issues
  - Architecture evolution
  - Architecture conformance
  - Frameworks, platforms, and ecologies
  - Conway’s law revisited
- Some questions to ponder

12 March 2010

Issue 1: Architecture Evolution

- Context: Increasingly, businesses must evolve their architectures
  - From A to C, through a series of incremental architectures B, B', B''
  - E.g., migrate batch-oriented systems to web-based interactive system; or migrate client-server system to service-oriented architecture (SOA).
- Issue: How do we approach this problem in a principled way?
  - Can we leverage past evolution histories?
  - How does this problem link to project planning, cost estimation, work assignments, etc?

12 March 2010
Issue 2: Architecture Conformance

- Context: We would like to make sure that the implementation conforms to architecture (and vice versa)
  - This is the Achilles Heal of software architecture
- Issue: What does it really mean to “conform” and how would we evaluate its satisfaction?

12 March 2010

Issue 3: Frameworks, Platforms, and Ecologies

- Context: We have been building on top of platforms and using software frameworks for most of the history of software engineering
  - This introduces an upward constraint on the architecture
  - The nature of such platforms has evolved
- Issue: What is the nature of modern platforms and how should architecture accommodate those?
  - Reflects a historical migration of architectural concerns
  - Needs to be rethought in the presence of the Internet

12 March 2010
Structure of the Mainframe Computer Industry

"Old" computer industry

DISTRIBUTION
Layer 5

APPLICATION SOFTWARE
Layer 4

OPERATING SYSTEM SOFTWARE
Layer 3

COMPUTER PLATFORMS
Layer 2

BASIC CIRCUITRY
Layer 1

Source: Intel

IBM DE C NCR WANG NEC

New Structure of the Computer Industry

"New" computer industry

DISTRIBUTION

Computer dealer
Value-Added reseller
Direct Mail
Mass
Other

APPLICATIONS

Spreadsheets
Lotus 1-2-3
Microsoft Excel
Borland's Quattro

Word processors
Graphics
Database

OPERATING SYSTEMS SOFTWARE

MS-DOS
Windows
Apple
Novell Netware
Banyan
IBM
Other

Unix OS/2

Computer platforms

IBM
Compaq
Other Intel-based personal Computers
Apple

PROCESSOR

Intel x86
Motorola 68K
Isa Other

Source: Intel

12 March 2010

Reprinted from The Economist, Feb 27, 1993
Issue 4: Conway’s Law Revisited

- **Context:** Conway’s “law” says that the structure of a software system reflects the structure of the organization that built it.

- **Issue:** What does this say about architecture?
  - Particularly an issue when we don’t build everything ourselves
  - Is organization-architecture conformance an attribute that we should pay attention to?

---

Some questions to ponder (for educators)

- Can we really teach people to be great architects?
  - .. If not can we teach them to be better architects?

- What is the role of domain knowledge?
  - .. Are we wasting our time teaching students general architectural principles?

- What kinds of assignments can get to the heart of the matter?
  - .. See workshops later today.

- Where are the great exemplars?
  - .. Perhaps Grady Booch has the answer.

- What can we learn from Google and Amazon?
  - .. Architectures for the new age?