

System Architecture

C15 - Engineering Document & Data Management Architectures

Tuesday, May 25th, 1999

10:45 - 12:00

Glen B. Alleman

Principal Consultant

Niwot Ridge Consulting

(720) 406-9164

glen.alleman@niwotridge.com

System Architecture

How good system architecture can be used to improve the chances of success for an EDM/PDM/ERP project.

No warranty for Carnegie Mellon and Software Engineering Institute Materials referenced in this presentation

NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED IN AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKE NO WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVELY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

3

Copyright Notice

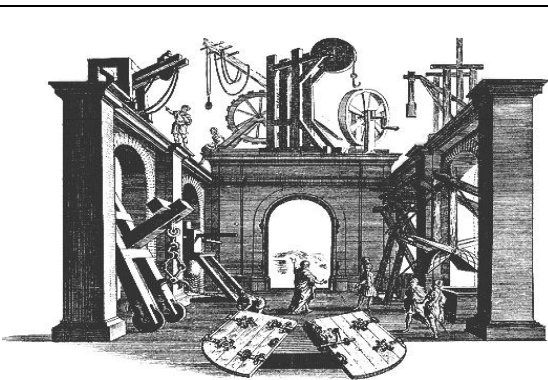
This document contains materials that are proprietary to Gateway Group, Inc. This work is protected as an unpublished work under the copyright laws of all countries that are signatories to the Berne Convention and the Universal Copyright Convention.

Copyright, © 1999 by Gateway Consulting Group, Inc., All rights reserved.
8610 Transit Road, East Amherst, New York 14051
(800) 668 2334

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

4



Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

5

What Is Architecture Anyway?

- Architecture is the set of decisions about any system that keeps its implementers and maintainers from exercising needless creativity.
- The architecture of a system consists of the structure(s) of its parts, the nature and relevant externally visible properties of those parts, and the relationships and constraints between them [d'sou99].

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

6

What Is Architecture at an Intuitive Level?

- Computing hardware architecture.
 - This is the view provided by the hardware vendors.
 - It consists of a small number of elements that can be scaled through replication.
- Network architecture.
 - There are two components, nodes and connections.
 - There are a limited number of topologies.
 - *The network is the computer* is the sun Microsystems logo.
- Building architecture.
 - Multiple views.
 - Architectural styles.
 - Engineering and materials based.

Some Warm Up Quotes

If a project has not achieved a system architecture, including its rationale, the project should not proceed to full scale deployment. Specifying the architecture as a deliverable enables its use throughout the deployment and maintenance cycles.

– Barry Boehm 1995.

Some Warm Up Quotes

I'm more convinced than ever. Conceptual integrity is central to product quality. Having a system architect is the most important single step towards conceptual integrity.

– Fred Brooks, *Mythical Man Month*, 1995

Some Warm Up Quotes

Architecture Lies at the heart of a successful system design.

How Can Architecture Be Placed in Context?

- *Requirements* definition and validation is concerned with the determination of the information, processing, and the characteristics of the system.
- *Architecture* is concerned with the selection of architectural elements, their interactions, and constraints.
- *Design* is concerned with the modularization and detailed interfaces of the design elements.
- *Implementation* is concerned with the representations of the data processing and data types that satisfy the design, architecture, and requirements [Perr92].

Manufacturing Issues

In the manufacturing and process industries, the unique requirements for integrating EDM, PDM, and ERP overwhelm the simple of buying off the shelf applications and plugging them together.

Two Contrasting Views of Architecture

- Architecture is a risk.
 - It will consume resources better directed at meeting a schedule.
 - It is too esoteric for this organization.
 - There are no bookable short term benefits.
 - A premature architecture can be more dangerous than none at all, since unproven architectural hypotheses turn into straightjackets.
- Architecture is a competitive advantage.
 - It lays the ground work for a commanding competitive advantage in the future.
 - It provides the mechanism for finally getting this place organized.
 - An immature architecture can be an advantage in a growing system because data and functionality can migrate to their natural places.

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

13

The Five R's of the Manufacturing Enterprise

- Produce the *right* product,
- with the *right* quality,
- in the *right* quantity,
- at the *right* price, and
- at the *right* time.

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

14

The Manufacturing Enterprise must:

- Leverage its core design and manufacturing competencies and pursue new business opportunities while outsourcing non-core activities.
- Implement new production strategies rapidly (mass customization, lean manufacturing, etc.) using the latest information technologies.
- Predict how change will effect the operational constraints of the business.

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

15

What Is the Problem Here?

There is a general disregard for architecture and the discussion of architecture. The results can be called the *big ball of mud* architecture. This architecture is constructed by expediency rather than design [root97].

Factors Driving the Increase in Complexity [Plac99]

- Exploiting new technologies such as Internet and intranet capabilities.
- Integration of solutions with a wide variety of legacy systems in the pursuit of improved business operations.
- The integration of multiple applications.
- The customization of applications to meet user business process requirements.

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

17

What's the Difference Between Good and Bad Architecture?



Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

18

What Is the Problem Here? [Barr95]

- Organizations have amassed enormous capital assets in equipment and software systems. The current systems have (or will have) paid for themselves. But, they behave like a child that has grown up but never left home.
- The methodologies currently being used are designed to define, acquire and build the systems of *the past* not *the future*.
- The information system's groups are structured to *increase productivity*. They are not structured to contribute to the strategic progress of the enterprise's product development and manufacturing activities.

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

19

What Is the Problem Here?

- As a result...
 - Projects have become complex. Burdened with the *backoffice productivity* paradigm in a world of manufacturing automation.
 - Paybacks are elusive when the benefits are changing.
 - Vendors provide generic product solutions targeted to model industries, not legacy production environments.
 - Steering committees are fickle, with demands being made from all quarters.
 - Users demand custom solutions to immediate problems, not strategic redeployment of their resources.
 - The planning horizon is dictated by the stakeholders – the future is *now*.

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

20

Why Obvious and Simple Solutions Don't Work.

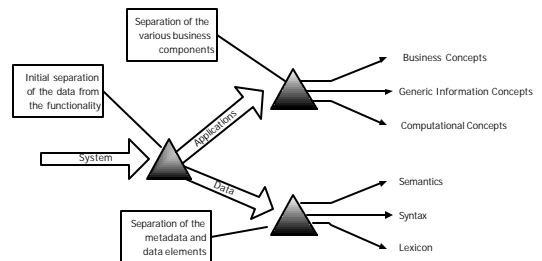
- The sequence of first approaches
 - Assemble a *best of breed solution* to meet specific needs
 - Customize a COTS product to fit the current requirements
 - Alter the business processes to fit the COTS solution
 - Roll you own solution
- Why these approaches don't work out of the box (OOB)
 - Systems can not be assembled out of parts
 - Customizing COTS product is risky business
- Sources of architecture failure
 - Failure to separate data and process
 - Failure to separate the concerns of the various system components

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

21

Failure to Separate Data and Process



Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

22

Failure to Separate Data and Process

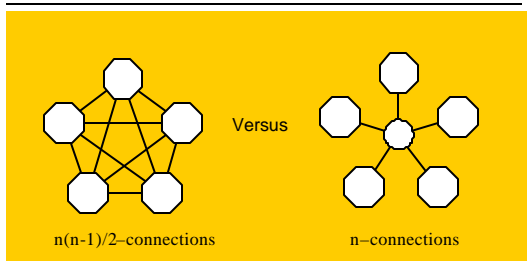
- If data is trapped inside an application:
 - Both the application and the data must be touched to make any alterations
 - The data cannot be reused
 - The semantics of the data is hidden and not available for reuse
 - New or different processes for the data are now restricted

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

23

Failure to Separate Concerns



Kalthoff, May 1999

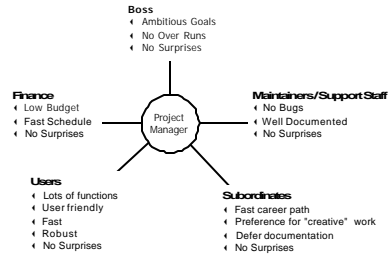
Copyright, Niwot Ridge Consulting, 1999

24

Failure to Separate Concerns

- Creates a tightly coupled system.
- Creates secondary effects during modifications to system components.
- Restricts rearrangement of the system component to match the business needs.
- Creates $n(n-1)/2$ connectivity management problem.

One Force No One Talks About



Who Should Care About Architecture?

- Architecture is the domain of IT strategy.
- IT strategy is the domain of the executive business decision process.
- *Strategy is the mechanism for creating fit among a company's activities.*
 - The success of strategy depends on doing many thing right, not just a few.
 - If there is not a fit among the things done right, then there is no strategy – just good tactics.
- Architecture is the foundation on which strategies are built.

What is Strategy?

- Improving operational effectiveness is a necessary part of management, but it is not strategy.
- Operational effectiveness involves continuous improvement of processes that have no trade-off's.
- Strategy involves the continual search for ways to reinforce and extend the company's position in the market place.

What is Strategy?

- Fit among a company's activities creates pressures and incentives to improve operational effectiveness.
- Fit means that poor performance in one activity will degrade the performance in others, so that weaknesses are exposed drawing management's attention.
- With increasing fit, improvements in one activity will pay dividends in other areas.

Mating Architecture and Strategy

- The challenge for the stakeholders is to create *fit* among the IT components and their matching business components.
- *Architecture is the mechanism for creating this fit.*

Creating Fit through Architecture

- Form, function, best use of resources and materials, human interaction, reuse of design, longevity of the design decisions, robustness of the resulting entities are all attributes of well designed systems [Alex79], [Lea93].
- Architecture is a set of rules that defines a unified and coherent structure consisting of constituent parts and connections of how those parts fit and work together.

By Allowing Architecture to...

- Avoid islands of information and process.
- Maximize component reuse.
- Neutralize system interface consequences.
- Lay a foundation for rapid construction of adaptive systems.

Creating Fit with Architecture

- System architecture becomes the *Business Driver* for achieving:
 - Streamlined business processes
 - Reduced information complexity
 - Liberation of data from with confines of the program
 - Enterprise-wide integration
 - Rapid evolution of new technologies

Who Should Care About Architecture?

- Executive Steering Committees
- IT Steering Committee's
- Business Unit Managers
- Plant Managers
- Business Process Stakeholders
- Chief Financial Officers
- *Any stakeholder that can influence or is influenced by the software system.*

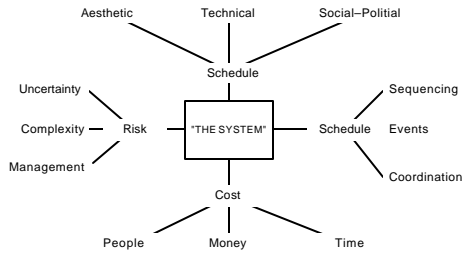
How Should These Stake Holders View Architecture?

- The System Architecture provides:
 - Communication Among Stakeholders – common abstraction of the system that can be used as basis for creating understanding.
 - Early Design Decisions – that enable priorities to be determined among competing concerns.
 - A Transferable Abstraction – constitutes a model for how a system is structured.
- The *representation* of the system as a set of architectural components is a critical success factor [Kazm99].

Addressing the Problem

How can the success rate for strategic IT projects be increased using good architecture practices?

Tensions on the System



Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

37

Guidelines for Constructing an Architecture-Based Strategy

- IT Project Principles
- System Architecture Principles
- Technology Assessment Principles
- Organizational Principles

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

38

Six Principles of Successful IT Projects [Dvor97]

- Make IT a business driven line activity, not a technology driven staff function.
- Make IT funding decisions like other business decisions.
- Drive simplicity and flexibility throughout the technology environment.
- Demand near-term business results from all IT acquisitions.
- Drive constant year-to-year operational productivity improvements.
- Build a business-smart IT organization and a IT-smart business organization.

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

39

Architecture-Based Processes

- Create a business case.
- Understand the requirements.
- Create or select the architecture.
- Represent or communicate the architecture.
- Analyze the architecture.
- Implement the system.
- Ensure the implementation conforms to the architecture.

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

40

Technology Assessment Principles

- Interoperability
- Platform migration
- Operating system, database, and application migration
- System scalability
- Network architecture
- Resource accessibility and naming
- Business resumption architecture
- Heterogeneous system federation

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

41

Organizational Principles

How the project and systems are organized depends on *context* of the business.

Organizational Principles

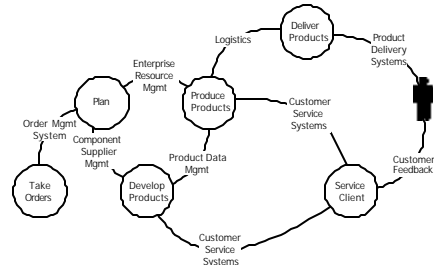
- Effective organizations do not always have the same specialized skills.
- All organizational alternatives have disconnects somewhere in their structure. The question becomes how to integrate the enterprise outside the boundaries of the organization.
- Continually expanding the boundaries of the organization in an attempt to absorb these disconnects will fail.
- Information architecture holds the key to providing true enterprise integration and must be decoupled from the organization chart.*

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

43

High Level Picture of a Manufacturing Organization



Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

44

Current Context

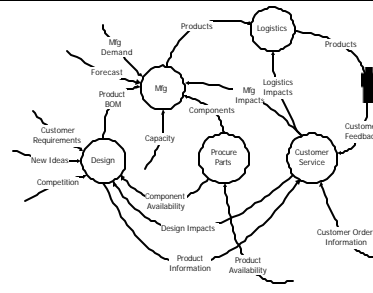
- ERP Systems
- PDM Systems
- EDM Systems
- Supporting Systems

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

45

Logical Connections

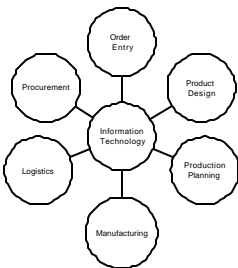


Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

46

A Glimpse into the Future [Srin98]



Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

47

Principles to Live By

- EDM Systems
 - Separation of metadata from the electronic vault indexing data
 - Isolation of published document formats from authoring document formats
 - Separation of workflow enablement from the user interface client application
 - Distribution of document indices as well as the documents
 - Separation of the software components into lifecycle layers:
 - Desktop
 - Middleware
 - Archival images
 - Keep all communications architecture *clean* don't mix paradigms

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

48

Principles to Live By

- PDM Systems
 - Separation of product data indexing schemes from document indexing schemes
 - Isolation of CAD file format dependencies
 - Isolation of workflow components
 - Availability of low footprint workstation
 - Isolation of PDM datamodels from other business process datamodels
 - Recognition that the ERP system owns the bill of materials, once the *approved for manufacturing* step has taken place
- *There is only one source for the BoM* is this the place?

Principles to Live By

- ERP Systems
 - Recognition that the ERP system owns the bill of materials, once the *approved for manufacturing* step has taken place
 - Isolation of the ERP datamodels from the external data models
 - Scalability of all data and processing across the enterprise
 - Across the network
 - Across multiple databases (with some form of replication, synchronization or two phase commit processes)
 - Isolation of the document management components from the document usage components

If This Is So Easy, Why Do Strategic IT Projects Fail?

In theory there is no difference between theory and practice. In practice there is.

Drivers for Commercial Off the Shelf (COTS)

- EDM Based COTS
 - Management of Change
 - Compound documents or objects
 - Publishing lifecycle management
- ERP Based COTS
 - Accepted data and processing models (SAP, Baan, Peoplesoft)
 - Accepted database enablers (Oracle)
- PDM Based COTS
 - CAD Integration (Metaphase, PTC)
- The Web Paradigm

Drivers for Customization

- Regulatory requirements
 - OSHA §1910.119
 - ISO 9000
 - EPA
 - Good Manufacturing Processes
- Legacy Data models
 - MRP
 - Logistics
 - Customer Service
 - Sales Support
 - Product configurators

Both of These Drivers Exist

- As well as,
 - CORBA Business Objects
 - Java
 - XML
 - *Open* systems
 - Shop floor and work cell automation
 - Autonomous devices

Why Projects Get In Trouble?

- No clear description of a functioning system.
- No framework in which to evaluate alternatives.
- No metrics by which to measure the alternatives.
- Too much emphasis on features and technology at the expense of business outcomes.
- Too much accountability for the technology and not enough accountability for the strategic consequences of the system.

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

55

No Clear Description of the Functioning System

- The *Boxology* approach to system architecture
 - Ranges from a simple circle diagram of physical layers to complex boxes with lines connecting them
 - What are these lines and boxes?
 - Hardware and software components
 - Processes
 - Business activities
- System architecture notations are available, formalized and proven in the field – Use Them

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

56

No Framework for Evaluation

- Without some framework to evaluate the architecture, the process of architecture-based management has little value.
- Choose some framework.
- Make it quantitative, with formal metrics.
- Measure decisions against the framework.

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

57

No Metrics for Evaluation

- Without a quantifiable set of metrics, all comparisons are *relative* and contain no meaningful data.
- Metrics include quantitative and well as qualitative measures.
- These metrics include:
 - Quality predictions [Carl92]
 - Cost and schedule [Boeh81]
 - Size and Performance

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

58

Too Much Emphasis on Technology

- Too much technology before the business problems have been identified.
- Once a technology has been selected, all the business issues must now be put into a confined framework.
- Technology will come later in the architectural development
 - Obviously some technology framework will be selected, but this should be avoided until it is no longer possible to proceed without understanding the technologies
 - Avoid at all costs adopting a specific vendor's technology too early in the process

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

59

No Accountability for the Business Outcome

- Someone has to own the system from a financial point of view.
- Some has to own the technical aspects of the system, deliver these on time and on budget and assure they meet the requirements.

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

60

How Does Architecture Help?

The Roman bridges of antiquity were very inefficient structures. By modern standards they used too much stone, and as a result, far too much labor to build. Over the years we have learned to build bridges more efficiently, using fewer materials and less labor to perform the same task.

– Tom Clancey, *The Sum of all Fears*

How Does Architecture Help?

- If You Think Good Architecture is Expensive, Consider the Cost of Bad Architecture
- The assumptions of the desired system are made explicit [Abow93]
- The components of the system should be made *orthogonal* to provide for reuse [Parn72]
- Proven techniques for bridging the gaps can be used if the architecture is understood [Yell94]
- Design patterns can be used as a source for architectural guidance [Gamm94]

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

62

What is Architecture?

- Architecture provides two primary roles:
 - Provides a level of abstraction through which the designers can reason about the system's behavior
 - Serves as the *conscience* for a system as it evolves
- Architectural descriptions are concerned with:
 - Systems structure – in terms of high level elements
 - Abstractions for interaction – between the system components
 - Global properties – in terms of the system's behavior

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

63

What Architecture is Not

- A Silver Bullet – only works on werewolves, sorry [Broo87].
- A Magic Potion – only works when the audience suspends all belief or is so distracted that they miss the *slight of hand*, sorry.
- Does not convert a bad concept into a good one.

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

64

How Does Architecture Make Money?

- Cost avoidance through standardization, business case deployment and strategic planning
- Enabling Information Technology deployment as a *strategic weapon*
- Resource utilization
- Agile *manufacturing* enabled through Information Technology

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

65

Critical Success Factors?

- A clear and concise specification of a working system
- A detailed user interface *prototype*
- A realistic schedule for all system deployments
- Explicit priorities of requests for applications
- Active risk management for both technology and personnel

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

66

Critical Success Factors, continued...? [McCon97]

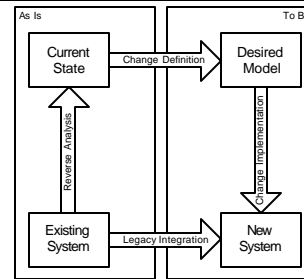
- A quality assurance plan
- Detailed activity lists
- A software configuration management program
- A clear and concise Software Architecture
- An integration plan

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

67

A Model for Identifying CSF's [Haum98]



Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

68

Active Risk Management

- *If you want a safe bet go to Las Vegas. The odds in Vegas are already fixed with a 97% pay out.*
- The management of risk is a critical success factor for any complex endeavor.
- The primary risk analysis topics are:
 - Schedule
 - Cost
 - Performance
 - Support
- Managing each of the risks within these areas is the responsibility of the Project Manager.

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

69

Software Architecture

- The software architecture is not the same as the system architecture.
- It is as important, but it is not the same.
- Without good software architecture, the integration of the system components is difficult at best and a disaster at worst.
- Imagine
 - Client / Server, embedded SQL statements, PL/SQL server components, *Flat File* database tables, a Web Browser user interface and Visual Basic as the integration environment?

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

70

Architecture Alternatives



How many ways can we rearrange the system, without changing the outcome?
 How many ways can we rearrange the deck chairs of the Titanic?

It Level Architecture Frameworks

- Zachmann [Zach87]
- CORBA [OMG91]
- SemaTech
- RM-ODP [ORMS97]
- 4+1 [Kurc97]
- Technical Architecture Framework of Information Systems (TAFIS) [DoD94]
- BOCA [Digr98]
- IBM OpenBlue
- The Open Group Architectural Framework

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

72

EDM Standards Used for Architecture

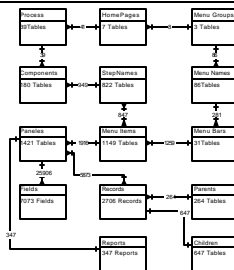
- WfMC – Workflow Management Coalition
- DMA – repository to repository specification
- WebDAV – standard for collaborative authoring on the Web
- ODMA – desktop to repository specification

Other Views of the World Masquerading as Architecture

- Enterprise Resource Management sold as infrastructure.
- Product Data Management sold as infrastructure.
- Electronic Document Management sold as infrastructure.
- Manufacturing Execution Systems sold as infrastructure.
- Client Server systems without an underlying framework.
- *The Web* in all its glory.
- Java Beans as a *verb*, *noun*, and *adjective*.
- Microsoft DNA.

A Simple View of an ERP System

[Aike99]



Blue Prints versus Principles

When the Titanic hit the iceberg, the Captain got out the blueprints, not the design principles.

What Happens When You Hit the Iceberg? [Boar98]

- Architectural representations are the models that allow us to understand the operational environment.
- In an ongoing enterprise there are models, either explicit, or implicit.
- Making the models explicit is a fundamental success factor for the enterprise.

Architectural Styles

- A set of components and types
- Topological layout of the system
- Semantic constraints on the system
- Connectors between the components

The Mother of all Architecture Wars

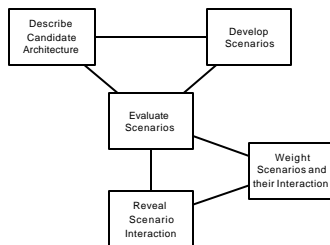
- CORBA versus DCOM
 - This is really Microsoft versus Not-Microsoft
 - Active-X is the current word for COM/DCOM, which was the previous word for OLE automation
 - It's a long complicated story, with lots of acrimony by all the parties
- The only real issues are, which approach works best in the business environment you have today and tomorrow?
- Leave all the technical mumbo jumbo to the technical boys and girls (at least for now)

How can we sort out all the alternatives?

The zeal for different opinions concerning religion, concerning government, and many other points...have in turn divided mankind into parties, inflamed them with mutual animosity, and rendered them much more disposed to vex and oppress each other than to cooperate for their common good

– James Madison

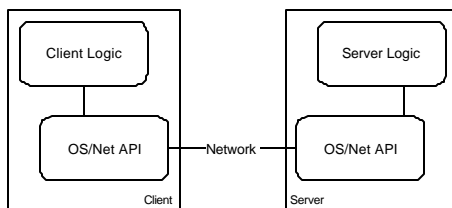
Evaluating Alternatives



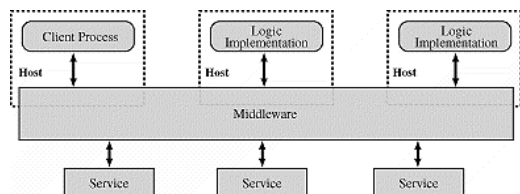
Now for the Pictures

The level of confusion at this point may be high, so lets step back and look at pictures for awhile.

Two Tier Architecture [Quoi98]

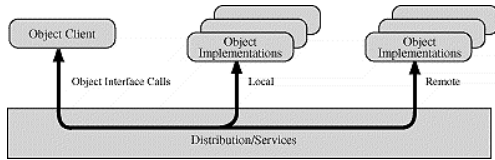


Three Tier Architecture [Quoi98]



Distributed Object Architecture

[Quoi98]

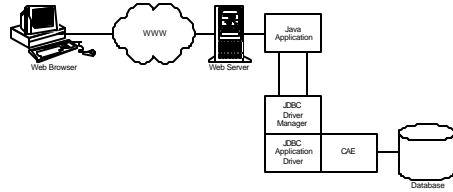


Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

85

Three Tier Web Architecture



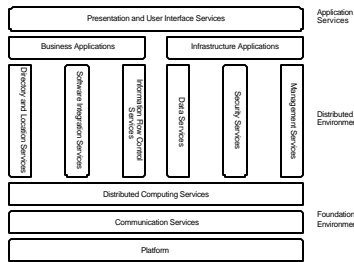
Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

86

The Open Group Architectural Framework

[Toga99]

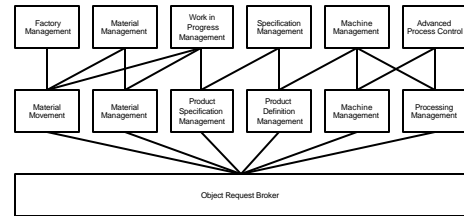


Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

87

SemaTech System Architecture



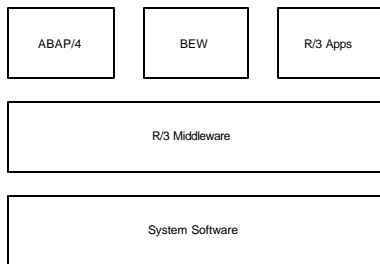
Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

88

SAP R/3 High Level

[Busk96]

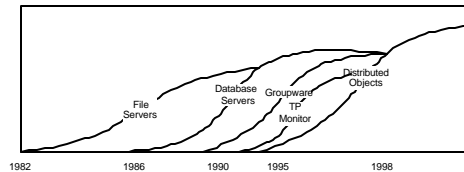


Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

89

Some Simple Taxonomy



Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

90

Where Does All These Lead Us?

- We need to understand the difference between *programming* architecture and *integration* architecture.
 - Java, Active-X Framework, Java Beans and the like are programming paradigms
 - CORBA is an integration paradigm, which can be implemented in a variety of programming paradigms.

Architecture as a Profession

- Professional societies and academic environments
- Some business environments that focus on architecture
 - Teleco's
 - Software vendors (operating systems, and the like)
 - Boeing
- Government laboratories
 - National Supercomputer Labs
 - National Center for Atmospheric Research
 - Los Alamos National Laboratory
- In these organizations architecture is a profession.
- How can we benefit from their work?

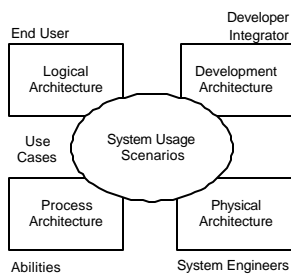
Components of the Architecture

- System Architecture
- Business Architecture
- Operational Architecture

System Architecture (4 + 1 View)

- Logical Architecture
- Development Architecture
- Process Architecture
- Physical Architecture

4 + 1 Architecture [Kurc97]



Logical Architecture

- Functional requirements of the business process that are directly implemented by the system
- Manual process that are not provided by the system
- Describes how the system *logically* functions

Development Architecture

- Vendor supplied architecture
- System integrator developed architecture
- The business process already in place (legacy architecture)
- The resource constraints (personnel, physical plant, geography, business culture)
- The *political* and *social* cultural constraints on the system

Process Architecture

- The *abilities* of the system that are needed to meet the business needs
- The business process flow that will exist after the system is deployed

Physical Architecture

- The computing infrastructure
 - Networks
 - Servers
 - Applications
 - Client Interfaces
 - Operational Resources

Physical Architecture Checklist

- Reliability
- Security
- Scalability
- Availability
- Manageability
- Interoperability
- Adaptability
- Affordability
- Ease of Use

Business Architecture

- Business Process Models
- Business Data Models
- Separation of Concerns
- Layered domains and their interaction

Operational Architecture

- Risk Analysis
- Project Management
- Personnel Management

Controlling the Architecture

Where does the control of the architecture rest?

Architecture, the ARB, and the IT Steering Committee

- How does the ARB, the architecture itself, and the IT Steering committee interact?
- Who is responsible *in the end* for the success of the system?
- How can all these (potentially conflicting) organizations work together for the common good of the end user, and ultimately the shareholder?

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

104

Making it Happen

- Take control of the current architecture
 - Information systems must be seen as assets
 - Information system inventory used to separate data from process
 - Enterprise standards used to control the options
 - Reduce the number of data sources
- Set priorities
 - Architectural dependencies defined to reduce rework
- Chart a course
 - Isolate and decouple legacy systems
 - Create a *federated* approach to solving the problem

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

105

Architecture Review Boards

- Reviewing an Architecture is *interesting*, but how can the architecture be controlled?
- This process is no different than building architecture control:
 - Architectural control committees in covenant controlled communities
 - City Planning Boards, with building codes and architectural guidelines
 - Client / architect relationships for commercial projects

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

106

Why Have an ARB?

- *...today's major problems with software are not technical, but management problems* [Broo87]
- Complexity of the Villain of all software system projects
 - When systems are interrelated, the total effort increases as $n(n-1)/2$ for n persons [Broo75]
 - Add to this the political, cultural, and business drivers
- *Some forum for defining and managing the system architecture is required for success*

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

107

Motivation for the ARB

- Architecture is the key to quality
- Analysis is only useful in the presence of clearly articulated goals for the system being analyzed:
 - The definition of *goodness* is mandatory
 - The earlier in the system's lifecycle the better

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

108

Role of the ARB

- Provide a forum for architectural discussions
 - Not the business processes
 - Not the vendor selection processes
 - Not the specific application programs
 - *But the architecture of the system and the facilities it delivers to the business*
- *If your going to drive to Cleveland it helps to know where Cleveland is*

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

109

Activities of the ARB

- Sorting out the differences between business goals and system architecture – *separation of concerns*
- Managing the Risk Assessment process
- Defining the outcomes for the system architecture
- Validating the business and system requirements against the selected architecture
- Managing the *Political* aspects of the system design process

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

110

Activities of the ARB

- Managing the *Political* aspects of the system design [Andr98]:
 - What is the project request before us today? Who wants it?
 - What is the projects purpose? What is the impact on profit, product development, customer retention, etc ?
 - What are the functional requirements?
 - What are the non-functional requirements?
 - Is the problem understood enough to prototype the solution?
 - If the prototype is acceptable, will everyone sign off?

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

111

In the end the ARB must...

- Recognize that specific goals must be met at each step of the project
- Recognize the importance of the stakeholders
- Emphasize people over technology
- Recognize business and organizational concerns
- Recognize contributions from other disciplines

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

112

Let's Not Get Too Enthusiastic Just Yet

There are plenty of Buzz Word traps to fall into before we wrap up here. One of the biggest, bestest, and magnificent buzz words is OBJECT-ORIENTED.

Let's Listen to some Vendor's

Smith's electronic snapple-fritz cellular phone is now Object-Oriented.



Our tools enable you to use the latest OO methods to solve your business problems.

We now have Object-Oriented mountain bikes...in stock, ready to go...



Yea Right!@#&

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

114

A Brief Overview of Objects [Camp98]

- What is an object?
- What are messages?
- What are classes?
- What is inheritance?
- So why do we care about any of this?

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

115

Objects are Closer Than You Think



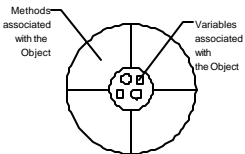
Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

116

What is an Object?

- An object is a bundle of variables and related methods
 - Everything the software knows (state) or can do (behavior) is expressed by the variables and methods.



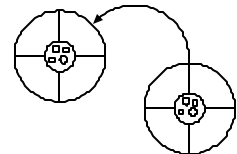
Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

117

What are Messages?

- Single objects are not very useful, so some form of communication between objects is needed.
 - This is done through messages
 - Messages have three components
 - Address of object
 - Name of the Method
 - Any parameters



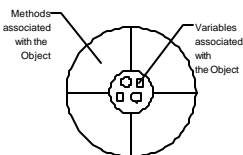
Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

118

What are Classes?

- A class is a blueprint or prototype that defines the variables and methods common to all objects of a certain type.
 - An object is an instance of a class
 - Objects provide modularity
 - Classes reusability



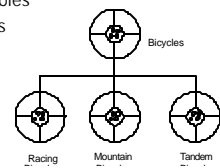
Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

119

What is Inheritance?

- Objects are generally defined in terms of classes
- Object oriented system allow classes to be defined in terms of other classes
- Each subclass inherits the variables and methods of the parent class
- The subclasses can override this inheritance in order to specialize the subclass



Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

120

Why Does Anyone Care About This?

- All of this would be of little interest if it were not for the tidal wave of Java and Smalltalk that is sweeping the mind share of the world.
- The old ways of thinking and doing things are just that old.
- Everyone speaks objects, draws objects, does object analysis, object design, they even dream objects.
- You may not agree with the way things are going, but you better get on the band wagon, or get out of the way.

Kalthoff, May 1999

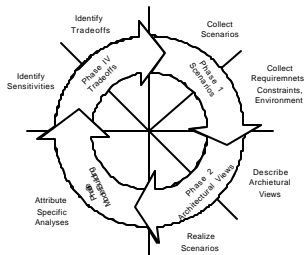
Copyright, Niwot Ridge Consulting, 1999

121

After That Brief Object Diversion

Let's get back to the problem at hand, organizing the architecture and the architectural review board.

Putting It All Together [Kazm98]



Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

123

One way to Set Up the ARB

- Software Architecture Analysis Method (SAAM) [Abow96]
 - Scenario development
 - Architecture description
 - Classification of scenarios
 - Individual evaluation of indirect scenarios
 - Assessment of scenario's interactions
 - Overall evaluation of architecture

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

124

Some Buzz Words to Focus on

- Integrated and flexible
 - What do these words mean?
 - Vendors use these words all the time.
 - There are the foundation of the marketecture
- The ARB's roles is to sort out the marketecture from the architecture.

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

125

The Cost and Benefits of the ARB

- Costs
 - Staff time
 - Organizational overhead
 - Consumption of senior content specialist
- Benefits
 - Financial
 - Forced preparation for the review process
 - Early detection of problems
 - Validation of requirements
 - Improved architecture

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

126

Questioning Techniques

- Scenarios
 - There is a strong desire to describe the system in terms of *abilities*
 - However, most software quality attributes are too complex
 - Scenarios provide a means of specifying attributes *in context*
- Questionnaire
 - Is a list of general and open questions that apply to all architectures
- Checklist
 - Is a more detailed set of questions that are developed after evaluating common sets of systems

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

127

Measuring Techniques

- Metrics
 - Are quantitative interpretations placed on an observable measurement
 - measuring techniques need to focus not only on the results, but also on the assumptions under which the techniques were used.
- Simulations, Prototypes, and Experiments
 - Prototypes help create and clarify the architecture
 - Simulations and prototypes may be an answer to an issue raised by a questioning technique

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

128

ARB Summary [Abow98]

- Have a formal review with external reviewers as a planned part of the project's lifecycle
- Time the review to best advantage. Consider an early architectural discovery review
- Choose an appropriate review technique
- Create a review contract
- Limit the number of qualities to be reviewed
- Make sure the review team includes an architectural expert, a domain expert, and support staff
- Insist on a system architect
- Collect scenarios and grow them into a check list

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

129

Some Sample ARB's

Gathering field data is a *sporty* business. Here are some samples from various sources. All the legal mumbo jumbo about the data is assumed to be known by the reader.

Sample ARB's

- There are not representative Architectural Review Board guidelines, since this is usually an *in house* process.
- The best guideline is found in [Adow98]. They outline three major techniques
 - Scenario based following the SAAM methodology
 - Questionnaire based using general and open ended questions about the architecture
 - Check list based using detailed questions developed after much experience in the domain.

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

131

Sample ARB's

- Some samples from simple research
 - Large western TelCo
 - Stanford University www.stanford.edu/group/APS/ARCinfo.html
 - University of Southern California <http://nunki.usc.edu:8082/~cs577/team8/>
- Some samples from personal communications

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

132

Sample ARB's Large Western TelCo



- Three Major Processes
 - Review Preparation / Consulting
 - Review Meeting
 - Issue Resolution
- Question List Oriented
- Review Handbook used to guide the architecture review
 - Handbook targeted at project managers
 - Methodology based review process (which is currently undergoing major rework)
- Reviewing is the primary tool for Quality Assurance

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

133

Sample ARB's Stanford University



- IT Management Principles used for managing EDM and financial systems migration
www.stanford.edu/group/APS/mgmprin.html
 - Purposes of Information Technology
 - Investment Criteria
 - Accountability
 - Information Availability
 - Data Collection and Administration
 - Infrastructure and Core Services
 - Application Design and Implementation

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

134

Sample ARB's University of Southern California



- **Digital Library Project**
 - **Architecture Review Board I**
 - This involves the review of LCO package developed. The review material should be put on the web a week in advance.
 - **Architecture Review Board II**
 - This review involves the analysis and review of the LCA package and is done by the client and CS 577b team members.
 - **Architecture Review Board III**
 - This is detail design review of the architecture to detect errors and find any priority capability left out in system detail design or problem with the system assumptions or interfaces.

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

135

Sample ARB's University of Southern California

- **Digital Library Project, continued...**
 - **Reviews and Inspections**
 - In this review, different Unit Test Completion Reviews (UTCR) are performed to check that each and every module of the system has been tested and meets the specified requirements.
 - **Transition Readiness Review**
 - The Software acceptance review (SWAR) is performed here by the client along with the development team.
 - **Release Readiness Review**
 - This RRR is performed at the end of the implementation phase of the project.

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

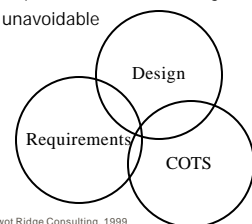
136

Architecture and COTS

Some final advice for deploying architecture based design, ARB's into the COTS environment.

Accept the Influence of COTS on System Design

- There are two common mistakes in the COTS business:
 - Designing the system without reference to a COTS application.
 - To allow one or more COTS products to dictate the design.
- COTS products have an unavoidable impact on the design of the system.



Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

138

Plan for Instability

- The Iron Clad Rule of the software business = *Things Change* and change very quickly
 - We're on Internet Time here boys and girls.
- It is usually the unstable technologies (Java, CORBA) that drive the technology forward, while creating the instabilities.
- Keep product-sensitive options open for as long as possible.
 - This is called *late binding*.
- Avoid *Vendor Lock* in pursuit of design stability.

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

139

Sustain Core Technology and Product Competency

- Deep product expertise is a critical success factor.
 - Good design decision about products cannot be made in the absence of sufficient knowledge about those products.
 - As the number of products increases so does the need for *spanning knowledge* for the interaction between products.
- Consultants provide a source for this deep knowledge.
- Formative evaluation techniques can be used to build *just in time* knowledge.
- The trick is to have sufficient knowledge to recognize a critical design issue.

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

140

Understand Vendor Lock and Vendor-Neutral Options

- The software market is driven by differentiation, not standardization.
- Insulating products provides stability.
 - These abstract interfaces reduce the interface capabilities to a common subset if not built using the proper tools.
- Avoid the *de facto* vendor locks
 - Assure every product has a viable competitor.
 - Isolate data from process.
 - Use product specific features in non-critical or discretionary parts of a system.

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

141

Use Business and Software Analysis in Design Decisions

- Managing tradeoff between commercial products and business requirements requires a deep understanding of the product as well as the business.
- The architect must make early design decisions with regard to product stability.
- The architect must understand the costs of acquiring product expertise versus the cost of making a poor decision.
- The for business acumen is most apparent when addressing the issue of *vendor lock*.

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

142

Time to Wrap Up



Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

143

What Have We Learned Here?

- Marketecture is not architecture.
- Architecture is a profession.
- Without some type of plan its hard to tell where you're going.
- Avoid the big mistakes:
 - Vendor Lock
 - Failure to separate data, process, and concerns
 - Follow a *Check List* approach in the beginning
- Look for working examples
- Take advice from others, but work out your own methods.
- Trust but verify.
- This is harder than it looks.

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

144

Questions to Ask in the Absence of Architecture [Zach99]

- What models are going to be built?
- How is quality going to be dealt with?
- How is integration going to be managed?
- How will change be managed?

The BIG Tradeoffs

- Short term versus long term options.
- Implementation versus integration.
- Point in time solutions versus infrastructure.
- Expense based approaches versus assets based approaches.
- Implementation optimization versus enterprise optimization.

That's all Folks



If we don't change our direction, we're likely to end up where we're headed – Chinese Proverb

Resources

In order to have any credibility, a person must have a library of knowledge upon which to draw when confronted with questions that can't be answered or when there is doubt in the mind of the audience.

Resources

- Much of the work on system architecture is available on the Web:
 - Software Engineering Institute – www.cmu.sei.org
 - Software Program Managers Network – www.spmn.com
 - Association for Computing Machinery – www.acm.org
 - Institute of Electrical and Electronics Engineering (Computer Society) – www.computer.org
 - Networked Computer Science Technical Reference Library – www.ncstrl.org
 - Worldwide Institute of Software Architects – www.wvisa.org
 - Cetus links to architecture – www.cetus-links.org/top_architecture_design.html
 - Software architecture technology guide – www.ast.tds-gn.lmco.com/arch/guide.html

Resources

- In order for the ARB to make informed decisions, a minimum set of knowledge is necessary. The following books should be read by all members of the ARB:
 - *The Art of System Architecting*, E. Rechtlin and M. W. Maier, CRC Press, 1996
 - *Building Enterprise Information Architectures*, M. A. Cook, Prentice Hall, 1996
 - *Managing the Software Process*, W. S. Humphrey, Addison Wesley, 1989
 - *Managing Risk*, E. M. Hall, Addison Wesley, 1998
 - *Requirements Engineering: A Good Practice Guide*, I. Sommerville, John Wiley, 1997

Resources

- At some point the technical issues must be addressed. These include the evaluation of the proposed system's *abilities*. The following books should be read and understood:

- *Making Hard Decisions*, R. T. Clemen, Duxbury, 1996.
- *Managing Multi-Objective Decisions*, M Mollaghasemi and J. Pet-Edwards, IEEE Computer Society, 1996.
- *The Art of Computer System Performance Analysis*, R. Jain, John Wiley, 1991.
- *The Practical Performance Analyst*, N. J. Gunther, McGraw Hill, 1998.

References

Knowledge is of two kinds; we know a subject ourselves, or we know where we can find information upon it

– Samuel Johnson

References

- [Adow95] – “Formalizing Style to Understand Descriptions of Software Architecture,” G. Abowd, G. Allen and D. Garland, *ACM Transactions*
- [Abow96] – “Recommended Best Industrial Practice for Software Architectural Evaluation,” G. Adowd, L. Bass, P. Clements, R. Kazman, L. Northrop, and A. Zaremski, *CMU/SEI-96-TR-025*.
- [Abow98] – “Architecture Reviews,” G. Abowd, L. Northrop, and A. M. Zaremski, *Software Architecture in Practice*, edited by L. Bass, P. Clements and R. Kazman, Addison Wesley, 1998.
- [Alex79] – *A Timeless Way of Building*, C. Alexander, Oxford University Press, 1979
- [Aust97] – “How to Manage ERP Initiatives,” R. D. Austin and R. L. Nolan, *Harvard Business School Working Paper*
- [Aike99] – “Reverse Engineering New Systems to Smooth Implementation,” P. Aiken and O. K. Ngwenyama, *IEEE Software*, March/April 1999, pp. 26–43.

References, continued...

- [Bass98] – *Software Architecture in Practice*, L. Bass, P. Clements and R. Kazman, Addison Wesley, 1998
- [Barr95] – “IS 2000: A new Role for Corporate IS Departments and People,” A. Barr, address given to *Software Development* conference, San Francisco, 1995. <http://www.scip.stanford.edu/scip/avsgt/>.
- [Broo75] – *The Mythical Man Month*, F. Brooks, Addison Wesley 1975.
- [Broo87] – *Report of The Defense Science Task Force on Military Software*, F. Brooks, Chairman, 1987.
- [Boar98] – *Constructing IT Blueprints*, B. Boar, John Wiley & Sons, 1998.
- [Camp98] – *The Java Tutorial: Object-Oriented Programming for the Internet*, M. Campione and K. Walrath, Sun Microsystems, 1998.
- [Busk96] – *SAP R/3 System: A Client / Server Technology*, R. Busk-Emden and J. Galimon, Addison Wesley, 1996.

References, continued...

- [Broo87] – “No Silver Bullets – Essence and Accident in Software Engineering,” F. Brooks, *IEEE Software*, April, 1997.
- [Boeh81] – *Software Engineering Economics*, B. W. Boehm, Prentice Hall, 1981.
- [Clem96] – *Coming Attractions in Software Architecture*, P. C. Clements, *CMU/SEI-96-TR-008*, Software Engineering Institute, Carnegie Mellon University, 1996.
- [Clem96a] – “Software Architecture: An Executive Overview,” P. Clements and L. Northrop, *CMU/SEI-96-TR-003*, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, February 1996.
- [Carl92] – “Software Measurement for DoD Systems: Recommendations for Initial Core Measures,” A. D. Carlton, Software Engineering Institute, *CMU/SEI-92-TR-19*.
- [Digr98] – “Business Object Component Architecture,” T. Digre, *IEEE Software*, Sept/Oct 1998, pp. 60–69.

References, continued...

- [Dunc96] – *A Guide to the Project Management Body of Knowledge*, W. Duncan, Project Management Institute, 130 South State Road, Upper Darby, PA 19082, 1996.
- [Dvor97] – “Six Principles of High-Performance IT,” R. E. Dvork, E. Holen, D. Mark, and W. F. Meehan III, *The McKinsey Quarterly*, Number 3, 1997. [Foot97] – “Big Ball of Mud,” B. Foote and J. Yoder, University of Illinois at Urbana-Champaign, September 1997.
- [Garl95] – “Architectural Mismatch or Why It's Hard to Build Systems Out of Existing Parts,” D. Garlan, R. Allen, and J. Ockerbloom, *Proceedings of the Seventh International Conference on Software Engineering*, April 1995.
- [Garl93] – “An Introduction to Software Architecture,” D. Garlan and M. Shaw, *Advances in Software Engineering and Knowledge Engineering*, Volume 1, World Scientific, 1993.
- [Kruc95] – “The 4+1 View Model of Architecture,” P. Kruchten, *IEEE Software*, 12(6), 1995.

References, continued...

- [Kazm96] – “Classifying Architectural Elements,” R. Kazman, P. Clements, G. Abowd, and L. Bass, *ACM SIGSOFT Symposium on the Foundations of Software Engineering*, 1996.
- [Laro98] – *Software Engineering Risk Management: Finding Your Path Through the Jungle*, Version 1.0, D. W. Karolak, *IEEE Computer Society*, 1998.
- [Perr92] – “Foundations for the Study of Software Architecture,” D. E. Perry and A. L. Wolf, *ACM Software Engineering Notes*, October, 1992, pp. 40–52.

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

157

References, continued...

- [Kemp98] – “Manufacturing’s Use and Abuse of IT,” R. -D. Kempis and J. Ringbeck, *The McKinsey Quarterly*, Number 1, 1998.
- [DoD94] – *DoD technical Architecture Framework for Information Management (TAFIM)*, Defense Information Systems Agency (DISA) Center for Information (CIM), Volume I (Concept), Volume II (Guidance), V 2.0 Reston VA, October 1992.
- [OMG91] – *Common Object Request Broker: Architecture and Specification*, Object Management Group, Framingham, MA.
- [Plac99] – “Enterprise Solutions Structure,” E. C. Plachy and P. A. Hausler, *IBM Systems Journal*, 38(1), 1999, pp. 4–11.

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

158

References, continued...

- [SPNM99] – *Software Program Managers Network*, www.spmn.com
- [d’Sou99] – *Objects, Components, and Frameworks with UML: The Catalysis Approach*, D. F. D’Souza and A. C. Wills, Addison Wesley, 1999
- [ORMS97] – “RM-ODP Part 1 (Overview), Part 2 (Foundations), Part 3 (Architecture),” <http://www.iso.ch:8000/RM-ODP>
- [Sowa92] – “Extending and Formalizing the Framework for Information System Architecture,” Soaw and J. Zachman, *IBM Systems Journal*, 31(3), 1992
- [Tock97] – “Introduction to Zachman Framework,” S. Tockey, *ORMSC/97-06-08*, Rockwell International, 1997

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

159

References, continued...

- [Adow93] – “Using Style to Understand Descriptions of Software Architecture,” G. Adowd, R. Allen and D. Garlan, *ACM Software Engineering Notes*, December, 1993.
- [Shei97] – “Adapting Zachman Framework to Business Objects,” R. Shelton, *ORMSC/97-06-09*, Open Engineering, 1997.
- [Kazm98] – “The Architecture Tradeoff Analysis Method,” R. Kazman, et al, *CMU/SEI-98-TR-008*, Software Engineering Institute, 1998.
- [Kazm99] – “Representing Software Architecture,” R. Kazman, *SEI Interactive*, 12/98, <http://www.sei.cmu.edu/interactive>
- [McCon97] – “Software’s Ten Essentials,” S. McConnell, *IEEE Software*, March/April 1997, pp. 143–144.
- [Andr98] – “The Politics of Requirements Management,” S. Andriole, *IEEE Software*, November/December 1998, pp. 82–84.

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

160

References, continued...

- [Haum98] – “Requirements Elicitation and Validation with Real World Scenes,” P. Haumer and K. Pohl, *IEEE Transactions on Software Engineering*, 24(2), December 1998, pp. 1036–1054.
- [Gamm94] – *Design Patterns: Elements of Reusable Object-Oriented Design*, E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Addison Wesley, 1994.
- [Yell94] – “Interface, protocols, and semi-automatic construction of software adapters,” D. M. Yellin and R. E. Strom, *Proceedings of OOPSLA ’94*, October, 1994.
- [Witt94] – *Software Architecture and Design Principles, Models, and Methods*, B. I. Witt, F. T. Baker, and E. W. Merritt, Van Nostrand Reinhold, 1994.

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

161

References, continued...

- [Lea93] – “Christopher Alexander: An Introduction for Object-Oriented Designers,” D. Lea, SUNY Oswego, <http://g.oswego.edu/dl/ca/ca.html>.
- [Foot97] – “Big Ball of Mud,” B. Foote and J. Yoder, University of Illinois at Urbana-Champaign, September, 1997.
- [Shaw96] – *Software Architecture: Perspectives on an Emerging Discipline*, M. Shaw, and D. Garlan, Prentice-Hall, 1996.
- [Spew92] – *Enterprise Architecture Planning*, S. Spewak, John Wiley and Sons, 1992.
- [Rech97] – *The Art of Systems Architecting*, E. Reichtin and M. W. Maier, CRC Press, 1997
- [Quoi98] – “Distributed Computing Architecture,” T. Burghart, Quoin, <http://www.quoininc.com>.
- [Srin98] – “The Changing Role of Information Technology in Manufacturing,” K. Srinivasan and S. Jayaraman, *IEEE Computer*, March 1999, pp. 42–49.

Kalthoff, May 1999

Copyright, Niwot Ridge Consulting, 1999

162

References, continued...

- [Zach87] – "A Framework for Information Systems Architecture," J. Zachman, *IBM Systems Journal*, **26**, Number 3, 1987.
- [Zach99] – "Life is a Series of Trade-Offs and Change is Accelerating," J. A. Zachman, *Data to Knowledge Newsletter*, Jan/Feb, 1999.