

	Behavioral View	Structural View
Conceptual Architecture (Domain)		
Logical Architecture (Domain)		
Evolutionary Architecture (Domain and Deployment View)		

Architecture Definitions

Dana Bredemeyer
 Bredemeyer Consulting
 Tel: (812) 335-1653
 Fax: (812) 335-1652
 Email: dana@bredemeyer.com
 Web: <http://www.bredemeyer.com>

Ruth Malan
 Bredemeyer Consulting
 Tel: (812) 335-1653
 Fax: (812) 335-1652
 Email: ruth_malan@bredemeyer.com

Introduction

- Definitions of architecture
 - System Architecture
 - Software Architecture
 - Technical Architecture
 - Enterprise Architecture

What: System Architecture Eb Rechtin's Definition

- **System**
 - "A system is defined ... as a set of different elements so connected or related as to perform a unique function not performable by the elements alone." p7
- **Architecture**
 - "The term 'architecture' is widely understood and used for what it is--a top-down description of the structure of the system."
 - *Systems Architecting: Creating and building complex systems*, Eberhardt Rechtin, Prentice-Hall, 1991



Copyright 2001 Bredemeyer Consulting
<http://www.bredemeyer.com>

How to Lead, How to Follow and How to Get Out of the Way
EAC Boston, March 2001 Slide 3

Other System Definitions

UML 1.3: A system is a collection of connected units that are organized to accomplish a specific purpose. A system can be described by one or more models, possibly from different viewpoints.

IEEE Std. 610.12-1990: A system is a collection of components organized to accomplish a specific function or set of functions.

Rechtin adds the following to his definition: "The whole is greater than the sum of the parts, that is, the system has properties beyond those parts." p28 (1991).

We prefer Rechtin's definition and emphasize that not only does the system perform a unique function, but it has unique characteristics or qualities that are inherent in the system and not just the parts (you couldn't put the turbocharged engine of a Dodge Stealth in an Escort and get the performance of the Stealth, for example).

Note on System Thinking

Thinking about systems, or from a systems perspective, has its roots in work by Buckminster Fuller and Russell Ackoff, among others. More recently, it has been recognized as a field in of itself, called "systems thinking", and has intellectual leaders such as Peter Checkland and Peter Senge (author of "The Fifth Discipline"). Eberhardt Rechtin's "Systems Architecting: Creating and building complex systems" (1991) remains one of the most important works in this domain, and has been highly influential in our approach to architecture.

Software Architecture Definition

Bass, Clements and Kazman, 1997

- “The software architecture of a program or computing system is the **structure** or **structures of the system**, which comprise
 - software **components**
 - the **externally visible properties** of those components,
 - and the **relationships** among them.”

• Bass, Clements, and Kazman. *Software Architecture in Practice*, Addison-Wesley 1997



Copyright 2001 Bredemeyer Consulting
<http://www.bredemeyer.com>

How to Lead, How to Follow and How to Get Out of the Way
EAC Boston, March 2001 Slide 4

Bass et. al.’s Comment on Their Definition

“By ‘externally visible’ properties, we are referring to those assumptions other components can make of a component, such as its provided services, performance characteristics, fault handling, shared resource usage, and so on. The intent of this definition is that a software architecture must abstract away some information from the system (otherwise there is no point looking at the architecture, we are simply viewing the entire system) and yet provide enough information to be a basis for analysis, decision making, and hence risk reduction.” Bass, Clements, and Kazman. *Software Architecture in Practice*, Addison-Wesley 1997

Other Definitions of Software Architecture

“... software architecture is a set of architectural (or, if you will, design) elements that have a particular form. We distinguish three different classes of architectural element: processing elements; data elements; and connecting elements.” Dewayne E. Perry and Alexander L. Wolf. “Foundations for the Study of Software Architecture”. *ACM SIGSOFT Software Engineering Notes*, 17:4, October 1992

Software architecture is “the structure of the components of a program/system, their interrelationships, and principles and guidelines governing their design and evolution over time.” Garlan and Perry, guest editorial to the *IEEE Transactions on Software Engineering*, April 1995

Software architecture encompasses the set of significant decisions about the organization of a software system

- selection of the structural elements and their interfaces by which a system is composed
- behavior as specified in collaborations among those elements
- composition of these structural and behavioral elements into larger subsystem

Booch, *Presentation at Software Developers Conference 1999*

Software Architecture Views

Conceptual Architecture

- Architecture Diagram, CRC-R cards
- *Focus: identification of components and allocation of responsibilities to components*

Logical Architecture

- Updated Architecture Diagram (showing interfaces), Interface specifications, Component specifications and usage guides
- *Focus: design of component interactions, connection mechanisms and protocols; interface design and specification; providing contextual information for component users*

Execution Architecture

- Process View (shown on Collaboration Diagrams)
- *Focus: assignment of the runtime component instances to processes, threads and address spaces; how they communicate and coordinate; how physical resources are allocated to them*

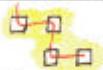
Presented by Malan and Bredemeyer at Comdex 98
<http://www.bredemeyer.com>

How to Lead, How to Follow and How to Get Out of the Way
EAC Boston, March 2001 Slide 5

Software Architecture Views

- *Conceptual Architecture*. The purpose of the conceptual architecture is to direct attention at an appropriate decomposition of the system without delving into details. Moreover, it provides a useful vehicle for communicating the architecture to non-technical audiences, such as management, marketing, and users. It consists of the Architecture Diagram (without interfaces) and an informal component specification (which we call CRC-R cards) for each component.
- *Logical Architecture*. The logical architecture adds precision, providing a detailed "blueprint" from which component developers and component users can work in relative independence. It incorporates the detailed Architecture Diagram (with interfaces), Component and Interface Specifications, and Component Collaboration Diagrams, along with discussion and explanations of mechanisms, rationale, etc.
- *Execution Architecture*. An execution architecture is created for distributed or concurrent systems. The process view shows the mapping of components onto the processes of the physical system. The deployment view shows the mapping of (physical) components in the executing system onto the nodes of the physical system.

Software Architecture Views

	Behavioral View	Structural View
Conceptual Architecture (abstract)	 Collaboration trace	 Architecture Diagram  Informal Component Specs (ORC-R)
Logical Architecture (detailed)	 Collaboration Diagrams	 Architecture Diagram with I/Fs  Interface Specs
Execution Architecture (Process View and Deployment View)	 Collaboration Diagrams showing processes	 Architecture Diagram showing Active Components

Copyright 2001 Bredemeyer Consulting
<http://www.bredemeyer.com>

How to Lead, How to Follow and How to Get Out of the Way
EAC Boston, March 2001 Slide 6

Structural and Behavioral Views

We use different views to enhance the understandability of the architecture and to focus on particular concerns separately.

- *Structural View.* If we accept that "architecture is the high-level structure of the system comprised of components, their interrelationships, and externally visible properties" (adaptation of the Bass, Clements, Kazman definition), the structural view is central. It consists of the Architecture Diagram, and Component and Interface Specifications.
- *Behavioral View.* In decomposing the system into components and designing their interfaces, we have to answer the question "How does this work?" Likewise, in understanding and using the architecture, we have to be able to answer the same question. This is the role of the behavioral view, with its Component Collaboration or Sequence Diagrams.

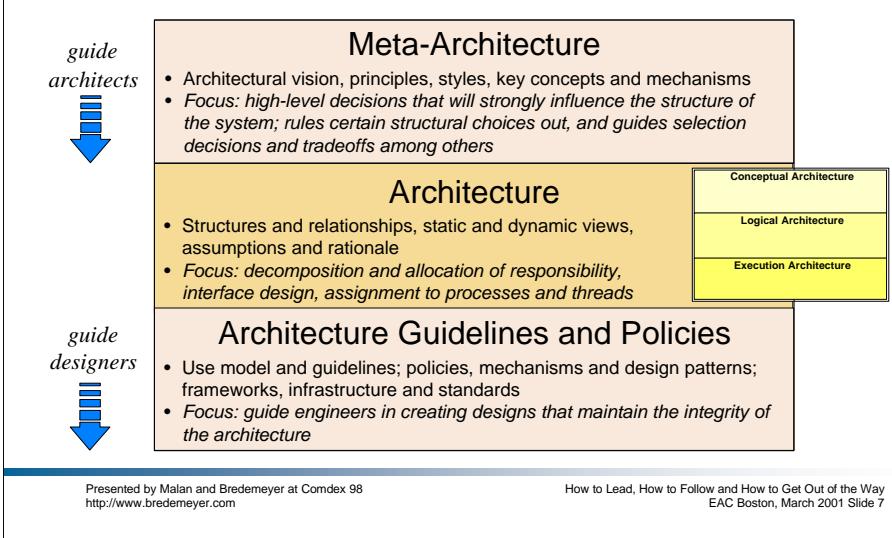
Architectural Model

An architectural model shows how significant properties of a system are distributed across its constituent parts. Coleman and Beringer, *Tutorial Presentation at UML World 2000*

Case Study

You can download the Common Clinical Context (CCOW) Architecture Specification from <http://www.mcis.duke.edu/standards/CCOW/Documentation.htm> at a "small" charge (of around \$75). This will provide you with examples of structural and behavioral views, and conceptual and logical architectures (though these names are not used).

Software Architecture Layers



Presented by Malan and Bredemeyer at Comdex 98
<http://www.bredemeyer.com>

How to Lead, How to Follow and How to Get Out of the Way
EAC Boston, March 2001 Slide 7

Background on the Software Architecture Model

Quite a number of top-flight architects created what they called “architecture”, but these were not the structure of any system, but more high level. In the case of HP JetSend, the architects articulated what they called an architectural vision (in terms of what the architecture would enable, rather than what it would look like), architectural principles (if you register as a developer on the JetSend site (<http://www.jetsend.com/JSHome.html>), you can access a wonderful example of the architectural principles, in this case for device interconnection), and key interconnection mechanisms. They did NOT identify or specify any components or interfaces. This was left to the architects of the wide variety of products that would use the JetSend “architecture”. However, if these product architects did not follow the guidelines and constraints established by the JetSend architects, then their architectures would not be JetSend architectures.

What: Technical Architecture

- The Open Group defines IT Architecture as
 - A formal description of an information technology system, organized in a way that supports reasoning about the structural properties of the system.
 - It defines the components or building blocks that make up an overall information system, and provides a plan from which products can be procured, and systems developed, that will work together to implement the overall system.
- (TOGAF FAQ)



Copyright 2001 Bredemeyer Consulting
<http://www.bredemeyer.com>

How to Lead, How to Follow and How to Get Out of the Way
EAC Boston, March 2001 Slide 8

IT Architecture

Boar defines IT Architecture as “a set of principles, guidelines and rules that guides an organization through acquiring, building, modifying and interfacing I/T resources throughout the enterprise.” (Boar, 1994).

Technical Architecture

See The Open Group Architectural Framework (TOGAF) for the Open Group’s IT Architecture Development Method (<http://www.opengroup.org/public/arch>). Note that it explicitly focuses on technical architecture (which they call IT Architecture). TOGAF explicitly outframes application architecture and data architecture.

IEEE Definition of Architecture

The IEEE Standard for Architectural Description of Software-Intensive Systems (IEEE P1471/D5.3) defines architecture as:

The fundamental organization of a system embodied by its components, their relationships to each other *and to the environment and the principles guiding its design and evolution.*

This definition may just as usefully be applied to technical architecture.

What: Enterprise Architecture Definition

- Enterprise architecture

- encompasses
 - Application/software architecture
 - Technical architecture (infrastructure such as networks and servers, middleware, etc.)
 - Information/data architecture
 - Organization/business architecture
- addresses enterprise-level objectives like
 - consistency, integration, interoperability, security
 - flexibility to make, buy or outsource IT solutions
 - reuse across applications (product lines or product families)



Copyright 2001 Bredemeyer Consulting
<http://www.bredemeyer.com>

How to Lead, How to Follow and How to Get Out of the Way
EAC Boston, March 2001 Slide 9

Enterprise Architecture

Enterprise architecture relates the constituent architectures (application/software, technical/IT, information/data and organization/business architectures), and provides the vision and guiding principles that govern all of these architectures.

At the level of the enterprise, the *Application Architecture* may be meta-architecture (providing guidance to system structuring, rather than the structure itself), or may be product line or product family architecture. This again may be just at a high level of abstraction (conceptual architecture) or detailed (logical architecture).

If a common platform is provided for reuse across a product family, then the architecture needs to be taken to the level of logical architecture. The common platform (shared infrastructure, components, frameworks, tools, etc.) may be viewed/managed as part of the *Technical Architecture*.