IEEE-Std-1471-2000
Recommended Practice for Architectural Description of Software-Intensive Systems

Rich Hilliard
rh@ConsentCache.com

14 November 2000

Overview

- What is IEEE 1471?
- History
- The IEEE 1471 framework: terms and concepts
- Requirements on Architectural Descriptions
- Applications of IEEE 1471
- Current Work and More Information

What is IEEE 1471?

- The IEEE Computer Society has developed IEEE-Std-1471-2000, *Recommended Practice for Architectural Description of Software-Intensive Systems has been developed*
- IEEE 1471 is a *recommended practice*
  - A "recommended practice" is one kind of IEEE standard
  - A using organization must decide whether to, and how to, employ IEEE 1471
- IEEE 1471 applies to *Architectural Descriptions*
  - Architectural Descriptions can be conformant
  - Systems, projects, processes or organizations cannot

History

- IEEE Architecture Planning Group:
  - First met August 1995, in Montreal
  - Final report to IEEE Software Engineering Standards Committee, April 1996
  - 6 Participants, 80 reviewers
- IEEE Architecture Working Group: May 1996 to December 1999
  - Bi-monthly meetings
  - 29 participants, 137 reviewers
- IEEE-Std-1471-2000, published in October 2000

IEEE Goals and Objectives

Chartered by IEEE *Software Engineering Standards Committee* to:

- Define direction for incorporating architectural thinking into IEEE standards
- Take a “wide scope” interpretation of architecture as applicable to software-intensive systems
- Establish a conceptual framework and vocabulary for talking about architectural issues of systems
- Identify and promulgate sound architectural practices
- Allow for the evolution of those practices as relevant technologies mature

Motivation: Why Architecture?

- Why do some systems “succeed”?
- Explicitly “architected” systems seem to turn out “faster, better and cheaper”
- Architecture is recognized as a critical element in the successful development and evolution of software-intensive systems

Scope of IEEE 1471

- *Software-intensive systems* are those complex systems where software contributes essential influences to the design, construction, deployment and evolution of the system as a whole
- There is a growing body of knowledge in the application of architectural concepts to these systems to attain the benefits of reduced costs and increased quality, such as usability, flexibility, reliability, interoperability and other system qualities

Organization of IEEE 1471

1. Overview
    - 1.1 Scope
    - 1.2 Purpose
    - 1.3 Intended users
    - 1.4 Conformance to this standard
2. References
3. Definitions and acronyms
4. Conceptual framework
    - 4.1 Architectural Description in context
    - 4.2 Stakeholders and their roles
    - 4.3 Architectural activities in the life cycle
    - 4.4 Uses of ADs
5. Architectural Description practices
    - 5.1 Architectural documentation
    - 5.2 Identification of stakeholders and concerns
    - 5.3 Selection of architectural viewpoints
    - 5.4 Architectural views
    - 5.5 Consistency among architectural views
    - 5.6 Architectural rationale
    - 5.7 Example use

A Bibliography

B Notes on terminology

C Examples of viewpoints

D Relationship to other standards

Using 1471

- IEEE 1471 is a *recommended practice*
  - One kind of IEEE standard
- 1471 applies to *Architectural Descriptions*
  - How to describe an architecture
  - *Not* a standard architecture, or architectural process, or method
- 1471 is written in terms of "shall," "should" and "may"
  - ADs may be checked for conformance to the recommended practice
  - 1471 does not define any conformance of systems, projects, organizations, processes, methods, or tools

What is an "Architecture"?

- Architecture: the fundamental organization of a system embodied in its components, their relationships to each other and to the environment and the principles guiding its design and evolution.

  where:

  - *fundamental organization* means essential, unifying concepts and principles
  - *system* includes application, system, platform, system-of-systems, enterprise, product line, ...
  - *environment* is developmental, operational, programmatic, … context of the system

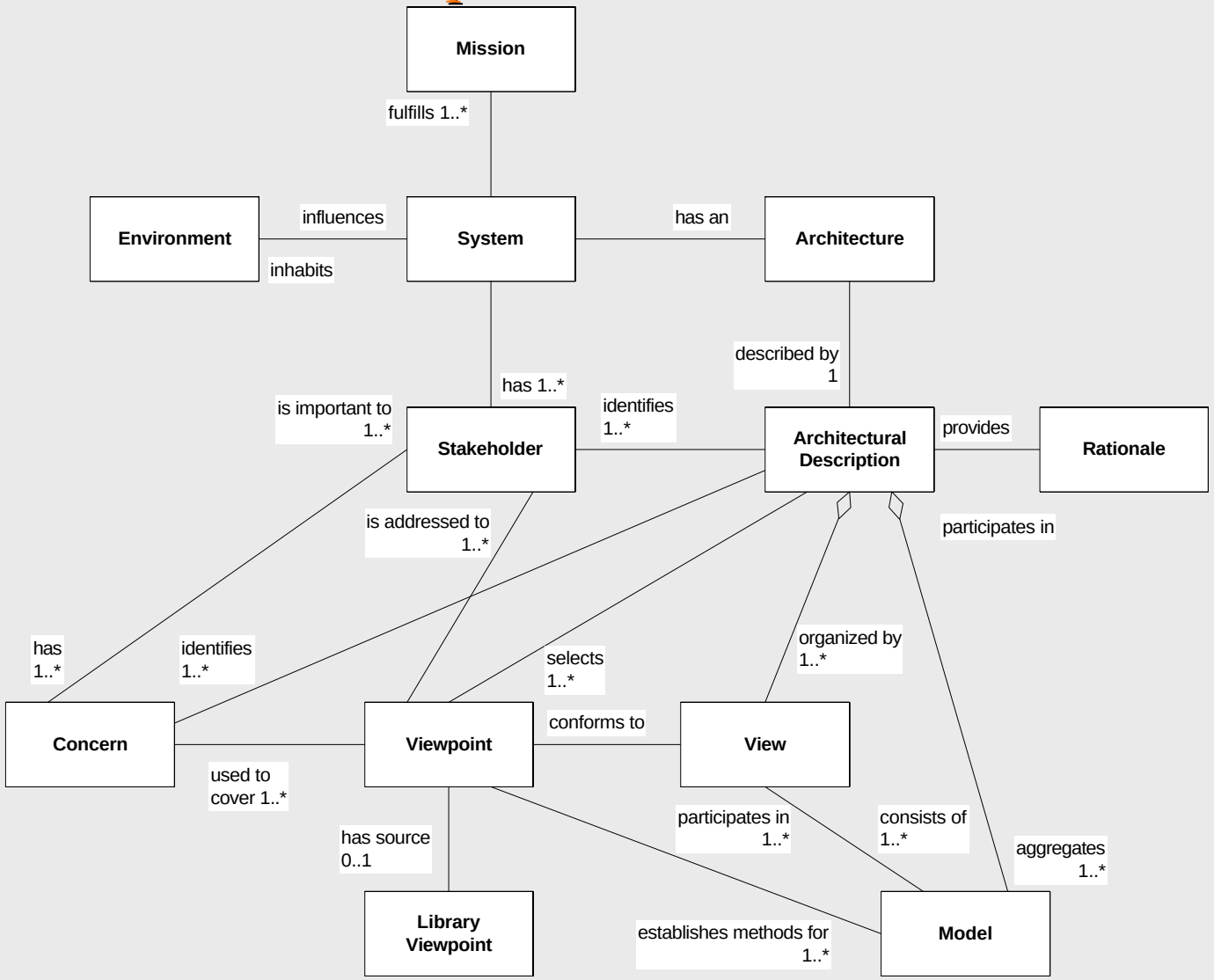
What is an "Architectural Description"?

- An *architectural description* (AD) is a collection of products to document an architecture
- IEEE 1471 does not specify the format or media for an architectural description
  - Notation-independent
- IEEE 1471 *does* specify certain (minimal) required content of an AD reflecting current practices and consensus

Role of Conceptual Framework

- To establish terms and concepts for architectural thinking
- To serve as a basis for evolution of the field where little common terminology exists
- To provide a means to talk about Architectural Descriptions in the Context of
  - System Stakeholders
  - Life Cycle
  - Uses of Architectural Description

IEEE 1471 Conceptual Framework



IEEE 1471 Requirements: Stakeholders and Concerns

- *ADs are interest-relative:*
  - An AD identifies the system's stakeholders and their concerns
- *Concerns form the basis for completeness:*
  - An AD addresses all stakeholders' concerns

Some Typical Stakeholders

- Client
- Acquirer
- Owner
- User
- Operator
- Architect
- System Engineer
- Developer
- Designer
- Builder
- Maintainer
- Service Provider
- Vendor
- Subcontractor
- Planner

IEEE 1471 Requirements: Views

- *Multiple views:*
  - An AD consists of one or more views
  - A *view* is a representation of a whole system from the perspective of a set of concerns
- *Views are themselves modular:*
  - A view may contain one or more architectural models, allowing a view to utilize multiple notations
- *Inter-view consistency:*
  - An AD documents any known inconsistencies among the views it contains

IEEE 1471 Requirements: Viewpoints

- *Views are well-formed*:
  - Each view corresponds to exactly one viewpoint
  - A viewpoint is a pattern for constructing views
  - Viewpoints define the rules on views
- *No fixed set of viewpoints:*
  - IEEE 1471 is “agnostic” about where viewpoints come from
- *Concerns drive viewpoint selection:*
  - Each concern is addressed by an architectural view
- *Viewpoints are first-class*:
  - Each viewpoint used in an AD is “declared” before use

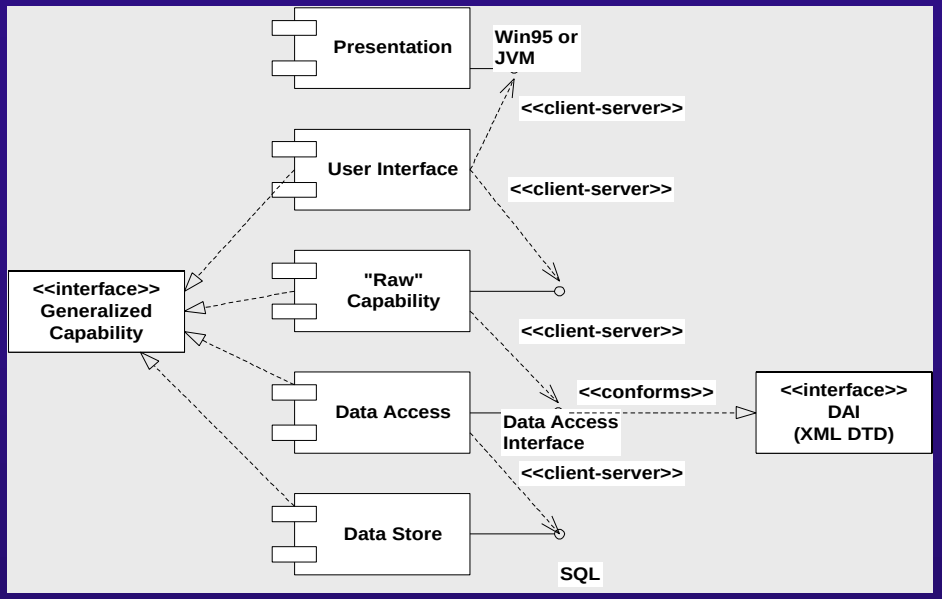
Declaring a Viewpoint

- *Each viewpoint is specified by*:
  - Viewpoint name
  - The stakeholders addressed by the viewpoint
  - The stakeholder concerns to be addressed by the viewpoint
  - The viewpoint language, modeling techniques, or analytical methods used
  - The source, if any, of the viewpoint (e.g., author, literature citation)
- *A viewpoint may also include*:
  - Any consistency or completeness checks associated with the underlying method to be applied to models within the view
  - Any evaluation or analysis techniques to be applied to models within the view
  - Any heuristics, patterns, or other guidelines which aid in the synthesis of an associated view or its models

A Viewpoint Example

- Viewpoint name: Capability
- Stakeholders:
  - The client, producers, developers and integrators
- Concerns:
  - How is functionality packaged?
  - How is it fielded?
  - What interfaces are managed?
- Viewpoint language
  - Components and their dependencies (UML component diagrams)
  - Interfaces and their attributes (UML class diagrams)
- Source: also known as Static, Application, Structural viewpoints

Example: Capability View



- The Capability View covers all system functionality for operating on data
- Capabilities are fielded using a 5-tier layered organization with interfaces between pairs of layers
  - Each layer is a capability
  - Entire stack is a deployable capability
- Capabilities can serve other capabilities

Library Viewpoints

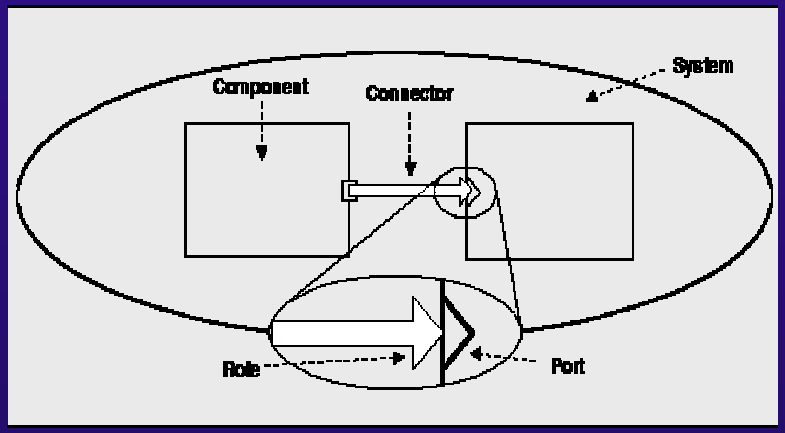
- Viewpoints are not system specific, unlike the stakeholders and views
- Hence, an active architect may be able to reuse viewpoint descriptions
- Equivalently, the viewpoints can be included by reference

IEEE 1471— What isn't Required

- No particular architecture description languages
- No required views or models
- No required formal consistency or completeness criteria

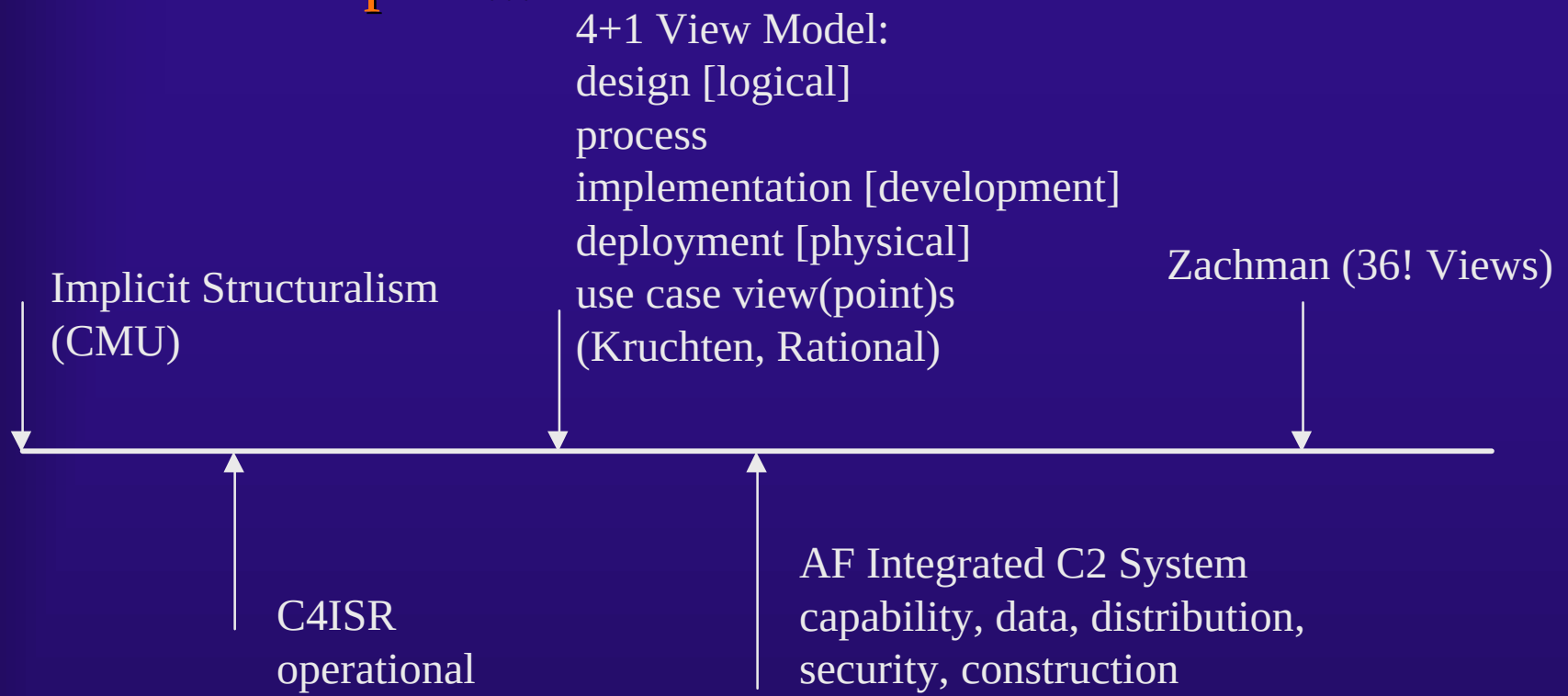
Example: Structural Viewpoint

- Concerns:
  - What are the computational elements of a system and their organization?
  - What element comprise the system?
  - What are their interfaces?
  - How do they interconnect?
  - What are the mechanisms for interconnection?
- Viewpoint language:
  - Components, connectors, ports and roles, attributes
- Analytic Methods:
  - Attachment, type consistency

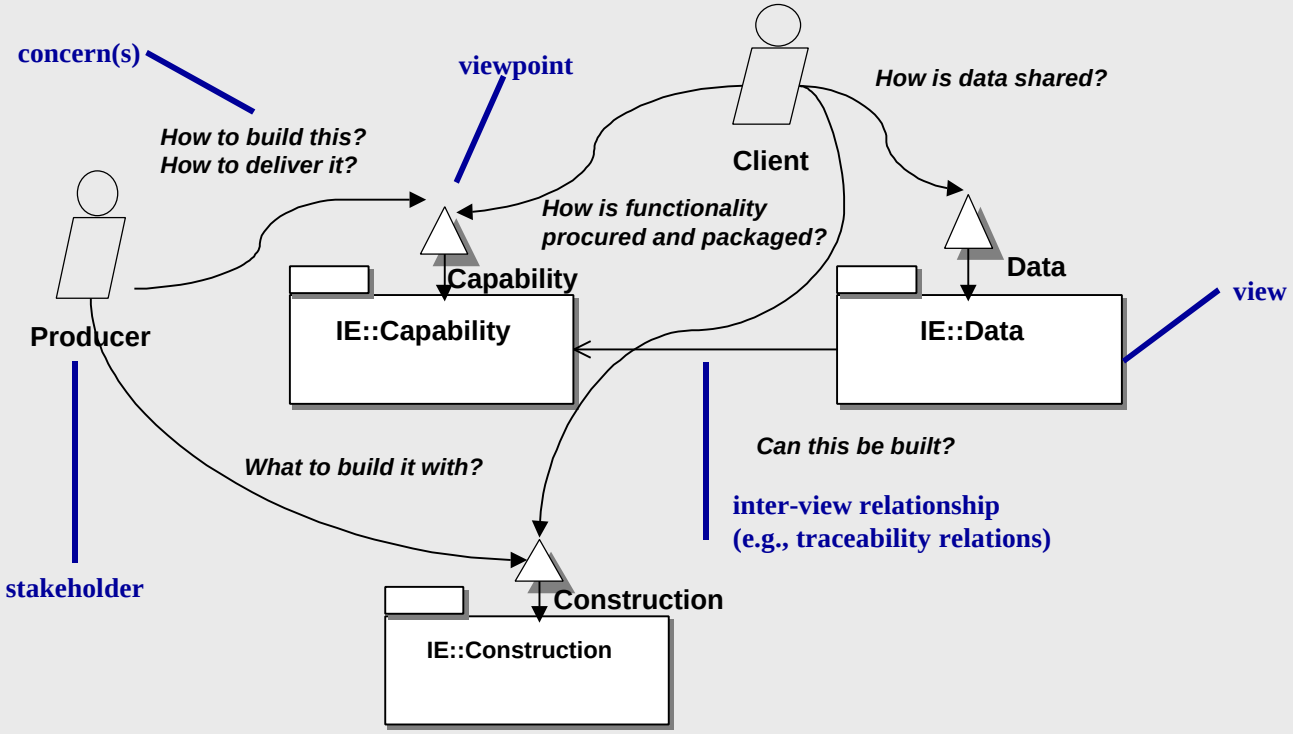


Based on: Acme: An Architecture Description Interchange Language, Garlan, Monroe, Wile, Proceedings of CASCON'97, November 1997

“Viewpoint Scale” 1471 is intended to encompass...



Organizing Viewpoints



Applications of IEEE 1471

- Architecture of Software-Intensive Systems Curriculum
- The Open Group Architecture Framework
- Software Architecture Review and Assessment (SARA) Industry Group
- Hewlett-Packard
- Rational
- Air Force Command and Control System Target Architecture

Current Status and Plans

- Current focus is on dissemination of the Recommended Practice via
  - Architect's curriculum and
  - Guide to the Recommended Practice

For More Information

- IEEE-Std-1471-2000 is available through IEEE Customer Service
  - +1-800-678-IEEE, or
  - http://standards.ieee.org/catalog/software4.html#1471-2000
- Visit the Architecture Working Group website:
  - http://www.pithecanthropus.com/~awg/