Knowledge mechanisms in IEEE 1471 & ISO/IEC 42010

Rich Hilliard
r.hilliard@computer.org

Two Themes

- Knowledge mechanisms in IEEE 1471 and ISO/IEC 42010
  - 2000 edition and on-going revision
- Toward a (bigger) picture of Architectural Knowledge (AK)

IEEE Std 1471™

- First formal standard for architecture description (2000)
- Now an international standard (2007)
- IEEE & ISO joint revision as ISO/IEC 42010 *Systems and Software Engineering — Architecture Description*

IEEE Std 1471 ™

- Built on an explicit ontology*
- Focused on descriptions not concepts
  - "the map is not the territory"
  - the blueprint is not the architecture

*Ontology, epistemology, meta model, conceptual framework, ...

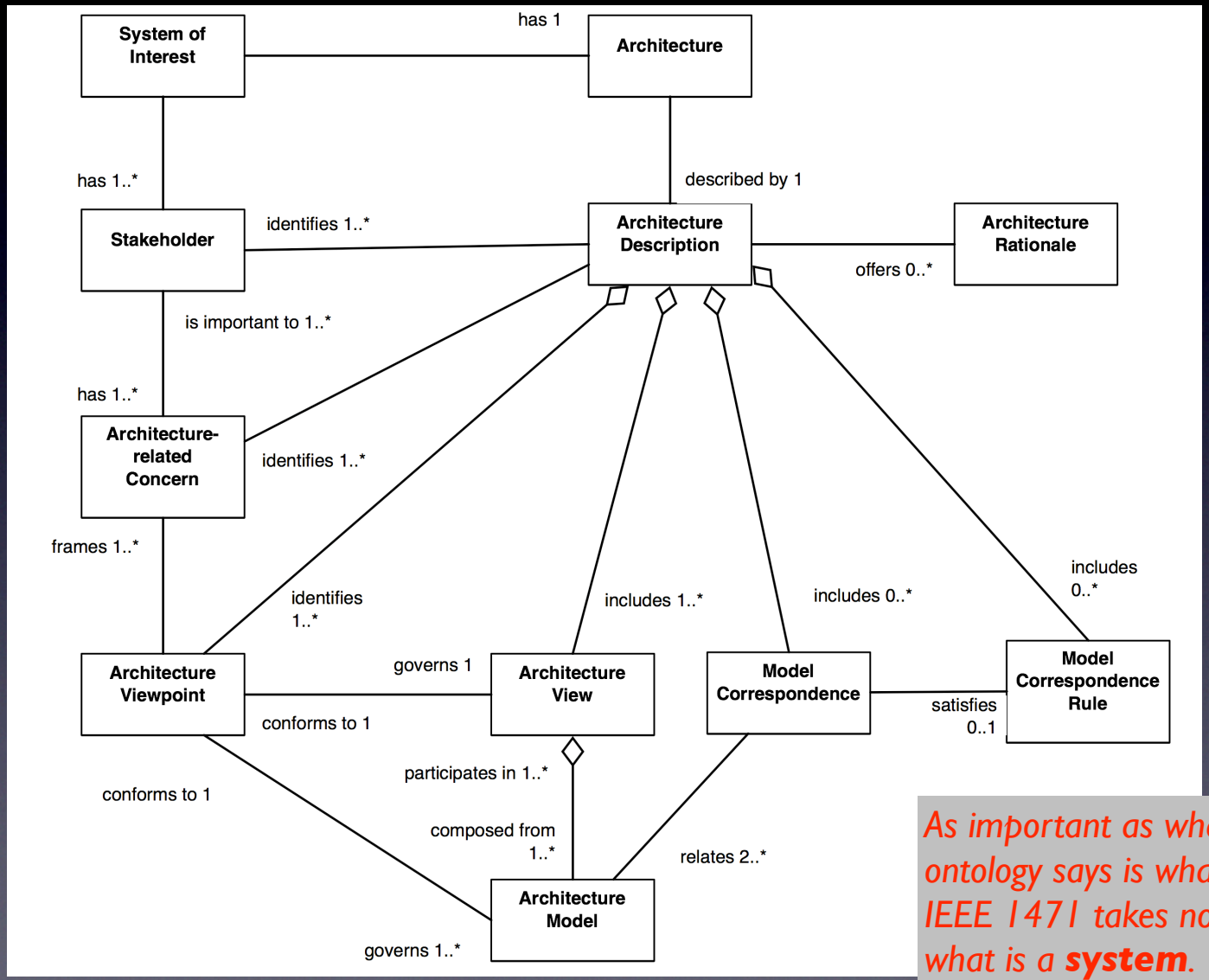
Knowledge mechanisms

- *knowledge mechanism*: a means of capturing knowledge
  - just as we distinguish Architecture from Architecture Description
  - let's distinguish *what we know* from *how we capture it*

Standards

- *Every standard is a knowledge mechanism*
- A standard reflects a community consensus, creating a filter on the world through its definitions and establishing rules on what to do when its definitions apply

Core Ontology



As important as what an ontology says is what it omits. IEEE 1471 takes no stand on what is a ***system****.*

Mechanisms

- (Architecture-related) System Concerns
- Stakeholders
- Views and Models
- Viewpoints and Model Types

System Concerns

- "area of interest in a system pertaining to developmental, technological, business, operational, organizational, political, regulatory, social, or other influences important to one or more of its stakeholders"

Separation of Concerns

"Let me try to explain to you, what to my taste is characteristic for all intelligent thinking. It is, that one is willing to study in depth an aspect of one's subject matter in isolation for the sake of its own consistency, all the time knowing that one is occupying oneself only with one of the aspects. We know that a program must be correct and we can study it from that viewpoint only; we also know that it should be efficient and we can study its efficiency on another day, so to speak. In another mood we may ask ourselves whether and if so: why, the program is desirable. But nothing is gained—on the contrary!—by tackling these various aspects simultaneously. It is what I sometimes have called "the separation of concerns", which, even if not perfectly possible, is yet the only available technique for effective ordering of one's thoughts, that I know of. This is what I mean by "focussing one's attention upon some aspect": it does not mean ignoring the other aspects, it is just doing justice to the fact that from this aspect's point of view, the other is irrelevant. It is being one- and multiple-track minded simultaneously.

— E Dijkstra, 1974

System Concerns: Examples

functionality, performance, reliability, security, information assurance, complexity, evolvability, openness, concurrency, autonomy, cost, schedule, quality of service, flexibility, agility, modifiability, modularity, inter-process communication, deadlock, state change, subsystem integration, data accessibility, distribution, persistence, safety, ...

Stakeholders (of a system)

- Individual, team, organization (or classes thereof) holding concerns with respect to a system

Role of Stakeholders and Concerns

- architecture: "fundamental conception of a system in its environment..."
- Stakeholders + Concerns = Environment

Viewpoints

- *viewpoint*: the conventions for constructing, interpreting and using a view
- A way of looking at a system

Specifying a Viewpoint

- concerns framed by the viewpoint
- languages, notations, model types used
- methods, heuristics, patterns, guidelines
- *new*: correspondences (with other viewpoints)

Viewpoints

- A Viewpoint is the *legend* for the map that is the View
- We "invented" viewpoints because we couldn't pick one set
- Inspired by Ross 1977, RM-ODP, Finkelstein et al.

Viewpoints à la Finkelstein et al.

Each viewpoint is composed of the following components, which we call slots:

- a representation style, the scheme and notation by which the viewpoint expresses what it can see;
- a domain, which defines that part of the "world" delineated in the style;
- a specification, the statements expressed in the viewpoint's style describing particular domains;
- a work plan, describing the process by which the specification can be built;
- a work record, an account of the history and current state of the development.

A. Finkelstein, et al., "Viewpoints: a framework for integrating multiple perspectives in system development," International Journal of Software Engineering and Knowledge Engineering, 1992.

Architecture models

- A view is composed of models determined by the viewpoint
- Models allow sharing between views

New mechanisms (proposed)

- models and model types: finer-grain reuse
- model correspondences and rules: linking views
- codifying architecture frameworks: for large-scale reuse and sharing
- rationale and decision capture

Model Correspondences

- In 2000 edition, we knew consistency between views is an issue but did not specify a mechanism
- Revision introduces
  - model correspondences and
  - model correspondences rules

Architecture frameworks

- *architecture framework*: conventions and common practices for architecture description established within a specific domain or stakeholder community
- Most architects work within a framework determined by their organization or client

Specifying an architecture framework

- a set of concerns
- typical stakeholders
- viewpoints
- model correspondence rules

Conformance

- An architecture description can conform
- An architecture framework can conform
- An AD can conform to a framework
- Proposed:
  - architecture viewpoint
  - architecture description language

Rationale and Decision Capture

Two AK Myths

- "Architecture descriptions are all about components and connectors"
- "Views don't capture decisions"

Two AK Myths

- Components and connectors are one possible viewpoint when using IEEE 1471
- Every view shows decisions, assumptions, constraints, ... based on the concerns it addresses

Rationale and Decisions

- Minimal treatment of rationale in 2000 edition
  - We've learned a lot since then, thanks to SHARK and others!
- Vague musing during IEEE 1471 development about decisions...

IEEE 1471 (early draft)

A very early draft of IEEE 1471 (draft 1.0, dated February 1998) contained a "Decision Viewpoint" that began:

5.3.8 Decision

The decision viewpoint documents the decisions about the selection of elements or their characteristics.

This viewpoint records the rationale for architectural choices. Typical models include:

- Mission utility
- Cost/Capability tradeoffs
- Element performance tradeoffs

Architect's Intent

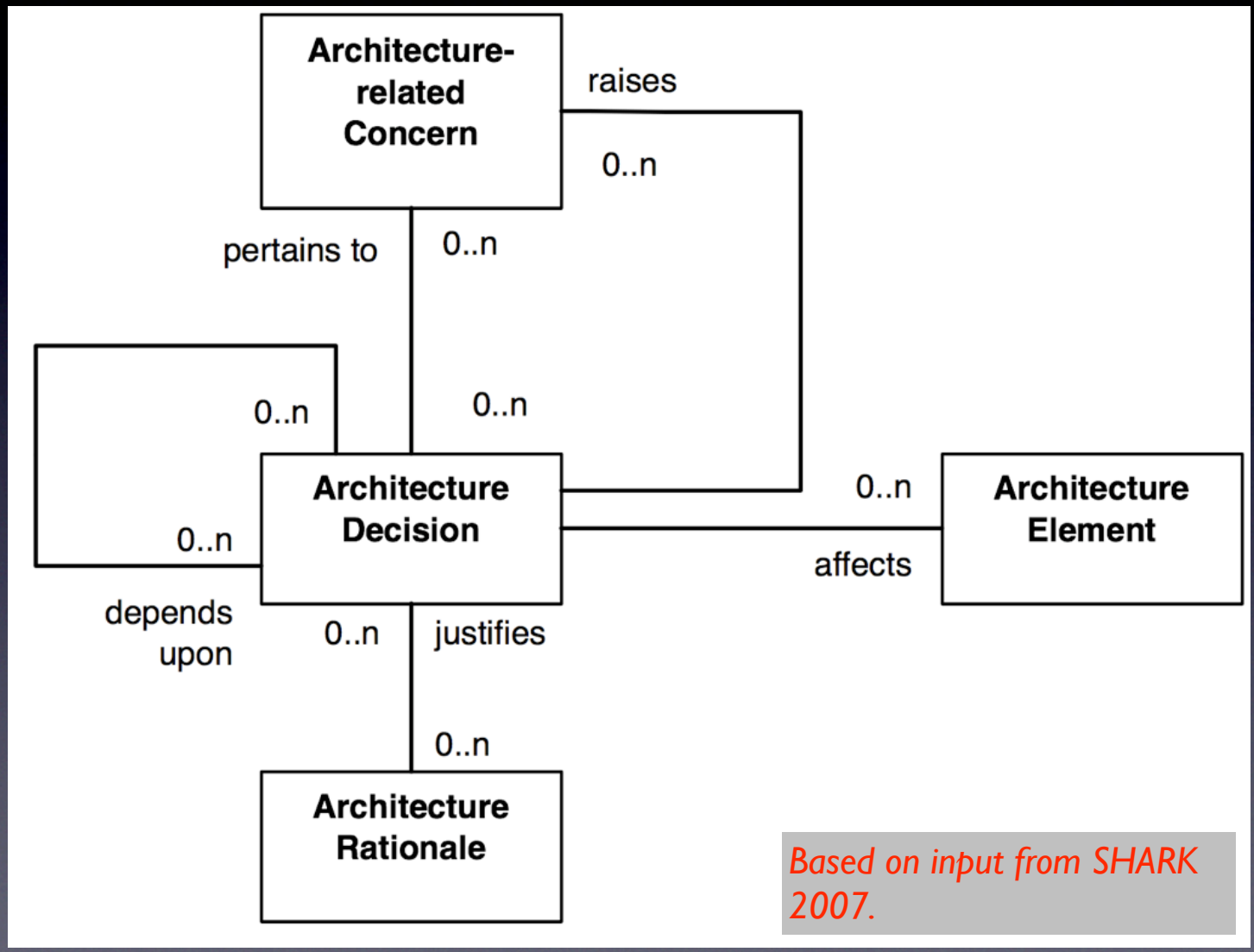
View Template: What readers need to know about each view

Purpose
Scope
Selected Viewpoint
Key needs
Assumptions
Key Decisions
    Commitments
Consequences
    Obligations and Freedoms
Open Issues

commitments:
decisions a designer is not at liberty to change
obligations:
lower-level decisions a designer must address
freedoms:
things left to the implementation

R. Hilliard and T. B. Rice, "Expressiveness in architecture description languages" *Proceedings of the 3rd International Software Architecture Workshop*, 1998.
A. Burns and M. Lister, "A framework for building dependable systems" *The Computer Journal*, 1991.
P.E. London and M. Feather, "Implementing specification freedoms" *Science of Computer Programming*, 1982.

Decision and Rationale in 42010



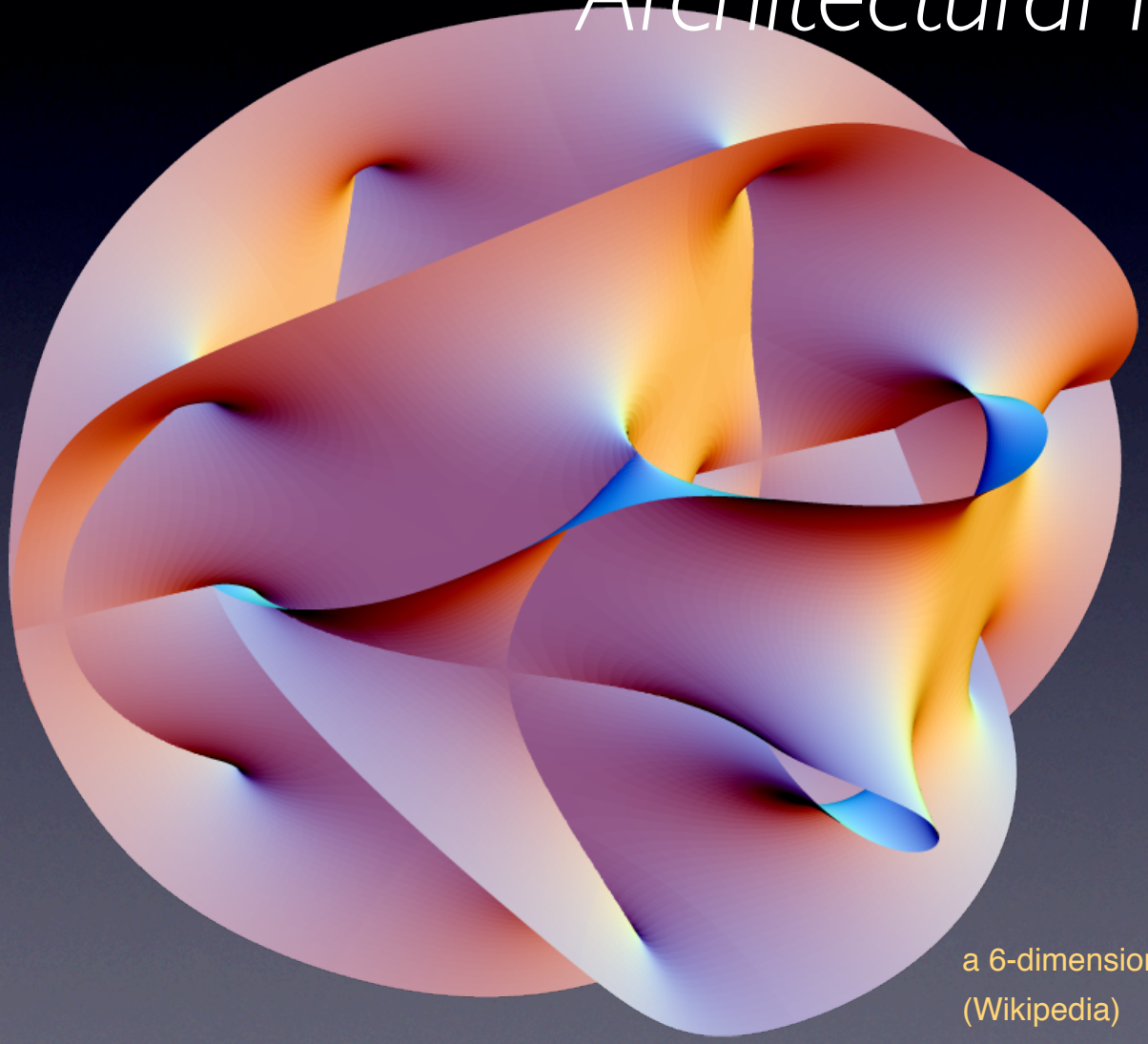
Based on input from SHARK 2007.

Styles of Decision Capture

- Annotations (as in Hilliard-Rice, 1998)
- Decision viewpoint: decisions are elements of the view with their relations (as in KCD*)
- Decision models: require each view to contain a decision model, relate elements of these models as in KCD

* Kruchten, Capilla, & Dueñas, "The Decision View's Role in Software Architecture Practice," *IEEE Software*, March/April 2009.

Toward a Bigger Picture of Architectural Knowledge



a 6-dimensional Calabi–Yau manifold
(Wikipedia)

Dimension: Levels

- System:
  - views, models, correspondence
- Organization, Community
  - viewpoints, model types correspondence rules,

Dimension: Areas of Interest

- System Concerns
- Disciplines, Domains, Implementation Technologies, ...

Dimensions: Social and Intentional

- Stakeholders have concerns
- Social:
  - actors, roles, duties, institutions, ...
- Intentions:
  - interested in, requires, needs, has as goal, decides, ...

Dimension: Forms

- Declarative (know that):
  - definitions, facts, principles, concepts, models, descriptions, artifacts, ...
- Procedural (know how):
  - strategies, techniques, methods, guidelines, ...

Challenge problem

The problem

- Styles, patterns and viewpoints: how are they the same? different?
- Compare and contrast as 3 mechanisms in active use for capturing architectural knowledge
- *Extra credit*: perspectives, view types

A theory of AK should offer insight into ...

- How are they the same?
- Are they interchangeable?
- What are differences?
- When to use each?
- Conditions on applicability?
- How do they interact, compose, interwork?

For more information on IEEE / ISO/IEC 42010

- Visit website, join users email group
- To participate in revision:
  - become an IEEE reviewer, or
  - join your ISO national member body

http://www.iso-architecture.org/ieee-1471/

Backups

Architectural Patterns

- The Name of the pattern
- The Problem which the pattern attempts to solve
- The Rationale provides a justification for the pattern
- The particular Context which the pattern solves a problem
- Forces (or tradeoffs)
- The Solution describes the structure and behavior of the result, and/or how to achieve that result
- Examples (and Visual Analogies) help explain the pattern
- Resulting Context (or, Force Resolution) explains what forces (issues and properties) the pattern leaves unresolved, and what other patterns might be applied to resolve these remaining issues

Source: "Gang of 4" book

Architectural Styles

- Vocabulary: What are the types of elements in the style? What relationships do they have? What are their properties? What are the rules of composition that determine how the vocabulary can be used?
- Semantics: What computational model do these elements support?
- Analyses: What forms of analysis are supported by the style?
- Implementation: What are the implementation strategies that allow one to produce an executable system?

Source: Clements et al., Views & Beyond book

Architecture Viewpoints

- Architectural concerns framed by the viewpoint;
- Stakeholders to be addressed by the resulting view;
- Resources: the model types, notations, language, modeling techniques, or analytical methods used;
- Associated operations: 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; and any heuristics, patterns, or other guidelines which aid in the synthesis of an associated view or its models

Source: ISO/IEC WD4 42010

What do we mean, architectural knowledge?

- Knowledge vs practice:
- Competence and performance:
- “Things” architects need to know