Software Architecture: Introducing IEEE Standard 1471

Mark W. Maier, Aerospace Corporation
David Emery, MITRE Corporation
Rich Hilliard, ConsentCache Inc.

Although architecture has become a popular term in the computing community, its use is inconsistent, often bearing little resemblance to the concept's origins in civil engineering. Architecture is used in various contexts to mean the instruction set of a central processor unit, the highest-level software modules in a large software system, or the overall structure of a business's information technology systems. In some contexts, the architecture is both the process and the outcome of specifying the overall structure, components, and interrelationships of a computer or a network. Other organizations speak of buying or acquiring an architecture. Architecture also describes a product line's shared attributes or features.

Despite these inconsistencies, there is a growing body of recognized practice in architecture as applied to computer systems. In 2000, the Computer Society approved IEEE Standard 1471, which documents a consensus on good architectural description practices. The decision process that led to IEEE 1471's approval demonstrates how standards can address conceptual issues and emphasizes the difficulties associated with resolving these kinds of issues in the standards development process.

Five core concepts and relationships provide the foundation for the approved IEEE 1471 version:

IEEE Standard 1471 identifies sound practices to establish a framework and vocabulary for software architecture concepts.

- Every system has an architecture, but an architecture is not a system.
- An architecture and an architecture description are not the same thing.
- Architecture standards, descriptions, and development processes can differ and be developed separately.
- Architecture descriptions are inherently multiviewed.
- Separating the concept of an object's view from its specification is an effective way to write architecture description standards.

IEEE 1471 focuses on both software-intensive systems and more general systems, such as information systems, embedded systems, systems-of-systems, product lines, and product families in which software plays a substantial role in development, operation, or evolution.

WHAT IS AN ARCHITECTURE?

Although defining architecture in the context of computing might seem like a simple task, it became one of the most contentious issues in developing the standard. This is not surprising considering that, despite approximately 5,000 years of practice, the civil architecture community has had little more success in precisely defining a building's architecture. The difficulty with this subject, which was the last issue resolved in the balloting process, is that it required elaborating on the concept that an architecture is a very complex property of a system rather than a thing itself.

Civil architecture influence

IEEE 1471 uses civil architecture as a metaphor for the design of software-intensive systems. The architect develops a limited representation of a building's physical structure, works with the client to understand the building's potential use and the client's resources for constructing the building, and determines the constraints that both the site and local laws place on what can be built. The architect is the client's trusted agent in coordinating all aspects of a building project, including the integration of structural, business, legal, and aesthetic concerns.

While the architect's role is broad, it does not extend to all of the building project's details. The architect's domain is the essential core, the aspects of the project that define usage, value, cost, and risk to within the client's tolerances. The architect's role is to help the client make the key decisions of when and how to go forward with the building program, and to define

the overall nature of that building program as it is contracted out to builders. This idea has been extended in common usage to more general things than buildings. The dictionary captures this idea by defining architecture as that which is essential or unifying about a system.

Definition debate

IEEE 1471 defines *architecture* as "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." This definition incorporates the idea that there is a difference between an architectural description and an architecture. An architectural description is a concrete artifact, but an architecture is a concept of a system. An architecture embodies a system's fundamental aspects. In this context, fundamental means the things that are important about the system as a whole, even if a single view cannot express that abstracted concept.

We must interpret what fundamental means in the context of both the stakeholders and the environment. The question is, "Fundamental to whom?" The architecture is more than just the overall structure of a system's components. A system usually does not have a top level in an arbitrary system/subsystem hierarchy.

Architectures and architecture descriptions

An architecture—a system's attributes—and what an architect produces—a set of documents—definitely are not the same thing. Similarly, different standards apply to architectures and architecture descriptions. We are familiar with this idea in civil practice. Communities have architectural standards that constrain how buildings are built, and they usually have standards for how architects draw building blueprints. There is a clear distinction between standard blueprint conventions and local building codes.

IEEE 1471 places normative requirements on architectural descriptions—the documents that describe a system's architecture. It does not standardize a system's architecture or the process for developing it. In this way, IEEE 1471 is more like a blueprint standard than a building code. It defines the equivalent of drawing and symbology conventions, although it does not define the full range of drawings needed to provide a description of any particular system.

IEEE 1471's most important elements are

- a set of definitions for key terms such as *architectural description, architectural view,* and *architectural viewpoint;*
- a separation of the concepts of architecture and architecture description to facilitate establishing standards for describing architectures (analogous to blueprint standards) and standards for constructing systems (analogous to building codes or zoning laws); and
- content requirements for describing a system's architecture.

By the same logic, IEEE 1471 does not standardize processes for producing an architecture description for two reasons. First, the standard's scope is descriptions, not process, and establishing a description standard doesn't require standardizing processes. Second, during the standard's development, it became clear that there is no community consensus on processes to compare with the consensus on architecture description concepts.

Views and viewpoints

An issue addressed early in the IEEE 1471 development process was whether to select one of the emerging architecture description languages as a standard. Because the architecture description concept has expanded to include budgets, business cases, and protocol definitions as well as physical component structure, it is evident that a single language is insufficient. Instead, we need to focus on another concept adapted from civil engineering—the notion that several views contribute to defining an architecture. We are accustomed to looking at building plans organized into several views—front, top, side, floor plan, plumbing, electrical, and so on—that meet the diverse needs of a system's various stakeholders.

In IEEE 1471, a *view* is a collection of models that represent one aspect of an entire system. A view applies to only one system, not to generalizations across many systems. The standard introduces the concept of *viewpoints* to capture common descriptive frameworks across many systems. Viewpoints are the vehicles for writing reusable, domain-specific architecture description standards. They establish the languages or notations used to create a view, the conventions for interpreting it, and any associated analytic techniques that might be used with the view.

An architecture description must define the viewpoint for each view it contains. This concept is consistent with emerging practices in a number of software fields in which viewpoints define representation standards for specific stakeholders, such as the International Organization for Standardization's *Reference Model for Open Distributed Processing.*

IEEE 1471'S ARCHITECTURE DESCRIPTION REQUIREMENTS

The new standard defines an architectural description's content requirements in terms of its elements. First, an architecture description must specify the system's stakeholders and identify their architectural concerns. Familiar concerns are

- *Functionality.* What does the system need to do?

- *Performance*. How will the system behave under heavy loads?
- *Security*. Can the system adequately protect user information?
- *Feasibility*. Can we implement the system?

Second, an architecture description must be organized into one or more views of the system's architecture. A view is not just any glimpse of the system—it must address identified stakeholders' concerns and it must be well formed. To provide a minimal completeness check, at least one view must address each identified architectural concern.

Finally, an architecture description must provide the rationale for making key architectural decisions. This can take the form of presentations of trade-offs the architects considered, alternatives they didn't choose, or other analyses that led them to choose the architecture that the architecture description documents.

IEEE Standard 1471 focuses on descriptions, not processes. By codifying existing consensus on key concepts and terminology, the standard provides a foundation that organizations can use for defining and applying sound architectural practices and serves as a basis for the evolution of future architectural approaches. ✳

Mark W. Maier is a senior engineering specialist at Aerospace Corporation. Contact him at mark.maier@ieee.org.

David Emery is a principal engineer at MITRE Corporation. Contact him at emery @davebert.mitre.org.

Rich Hilliard is chief technical officer at ConsentCache Inc. Contact him at rh@ ConsentCache.com.

Editor: Gary S. Robinson, EMC, 228 South Street, Hopkinton, MA 01748-9103; robinson_gary@emc.com