Introducing IEEE Standard 1471: *Recommended Practice for Architectural Description for Software Intensive Systems*

Mark W. Maier, The Aerospace Corporation

David Emery, MITRE

Rich Hilliard, ConsentCache, Inc.

IEEE Standard 1471 is the *Recommended Practice for Architectural Description for Software Intensive Systems*, developed by the IEEE's Architecture Working Group (AWG) under the sponsorship of the Software Engineering Standards Committee of IEEE. The draft *Recommended Practice* was produced between 1995 and 1998 by a group of approximately thirty participants, and over 140 international reviewers. In 1999 it went to ballot and was approved by the IEEE Standards Board in September of 2000.

This article reviews the concepts of IEEE 1471, and the rationale for their selection. Since a Recommended Practice is, by definition, partially a work-in-progress, we use the style of a FAQ for the article.

1. What were the Goals for IEEE 1471?

IEEE 1471 was developed in response to the recent and widespread interest in software architecture. The goals set for the Recommended Practice were:

1. Focus on software-intensive systems, but don't preclude more general systems whenever possible. This includes computer-based systems ranging from software applications, information systems, embedded systems, systems-of-systems, product lines and product families; any system where software plays a substantial role in the development, operation, or evolution of a system.
2. Establish a framework and vocabulary for architectural concepts. Despite the widespread interest in architecture in both the systems and software engineering communities, there has been wide diversity in vocabulary and organizing frameworks for essentially similar concepts. There are no agreed-upon definitions for terms commonly used in those communities such as "architecture," "architectural description," and "view," even though the terms are widely used and there is a fair agreement on what they mean.
3. Identify and promulgate sound architectural practices. Given the wide range of existing software and systems architectural practices, it is a goal of IEEE 1471 to provide a basis on which these practices may be defined, contrasted and applied.
4. Choose elements to standardize that facilitate the continued evolution of architectural technologies. Architectural practices are rapidly evolving, both in industrial use and in the research arena, with respect to architecture description languages, architectural methods, analysis techniques, and architecting processes. It is hoped these practices can be communicated, documented and shared via the framework of IEEE 1471.

2. How is IEEE 1471 Used?

IEEE 1471 places a set of normative requirements on Architectural Descriptions (AD), which are documents produced to describe a system's architecture. These normative requirements are deliberately incomplete; each implementing organization must extend the framework of IEEE 1471 to make its own AD standards. In this sense IEEE 1471 is similar to a standard for blueprints. It defines the equivalent of

drawing and symbology conventions, although it does not define the full range of drawings needed for an adequate description on any particular system. The most important elements of IEEE 1471 are:

1. A normative set of definitions for terms including: architectural description, architectural view, and architectural viewpoint
2. A separation of the concepts of "architecture" and "architecture description" to facilitate separating standards for how architectures are described (analogous to blueprint standards) from standards on how systems should be constructed (analogous to building codes or zoning laws).
3. A conceptual framework for how these concepts are implemented in the context of the many uses of architectural descriptions
4. A set of normative requirements on the elements of an architectural description of a system and the relationships among those elements

3. What are the main contributions of IEEE 1471?

In the context of current work on software-intensive systems architecture, IEEE 1471 codifies several practices for the first time. These are:

- Architectures of systems are to be considered in the context, or *environment*, of the system of interest.
- Principles for architectural description are separated from principles of architectures (e.g. of a family of related systems). As such, IEEE 1471 is a standard for architectural description—not on how to build or evaluate the resulting architectures.
- Architectural descriptions are constructed from multiple viewpoints, with each corresponding view covering an identified set of system concerns.
- Architectural descriptions are structured to meet the needs of the various stakeholders of the system, including, but not limited to, the builders of the system.
- Minimal well-formedness criteria on views are established by requiring their association with documented *viewpoints*, or templates for those views.

These ideas are discussed further below.

4. How does IEEE 1471 define architecture?

Interestingly, this issue became one of the most contentious in the development of the standard. The flip answer is that we don't know, but we know one when we see one. More seriously, it became clear in the discussions how difficult it is to provide a really good definition of the concept. Perhaps this is not surprising, given that the civil architecture community, with 5000 years or so of practice, has had little more success in precisely defining the architecture of a building. This difficulty in defining the concept of "architecture" has not, it is important to note, been an impediment to defining standards for what architects produce or for standardizing the types of models they use. The concept is hard to make precise, even when it is clear how to make use of the concept in development.

Broadly speaking, architecture of a system is that which is essential or unifying about a system. It is that set of system attributes that largely determines the system's value, cost, and risk. The definition in the recommended practice is:

> **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.

There are several key ideas in this definition. First, architecture is a conceptual attribute of a system, whereas an architectural description is a concrete artifact, representing an architecture. The architecture itself may or may not be written down. The architectural description is not the architecture, it is a description of something which is fundamentally conceptual.

Second, architecture embodies “fundamental” things about a system. Here we mean “fundamental” in the sense of an abstraction of things that are important about the system as a whole, even if that abstracted concept cannot be expressed in a single view. "Fundamental" must be interpreted in the context of stakeholders and the environment. We cannot know what is fundamental about a system without knowing “fundamental to whom?” There are some things that architecture definitely is not. Foremost, architecture is *not just* the overall structure of physical components that make up a system. While physical structure is often a fundamental aspect of a system, it is not always a fundamental aspect. “Fundamental things” *mustn't* be interpreted in the sense of the top level in some arbitrary system/subsystem hierarchy. In general, a system will exist in many such hierarchies and no single one can be preferred.

Third, an architecture exists in context, not in the abstract. To understand a system's most fundamental characteristics (i.e., architecture) we must understand how the system relates to and is embedded into its environment.

5. What does conforming to IEEE 1471 mean?

An architectural description is conformant with IEEE1471 if it meets the requirements in clause 5 of the Recommended Practice. These requirements pertain to the documentation of the architecture, identifying the key stakeholders for the architecture, factoring the documentation into views, the existence of well-defined techniques for documenting each of those views (called *viewpoints*), insuring consistency between views, etc.

6. Why does IEEE 1471 have both views and viewpoints?

The two concepts are different, and both are important. A *view* is an aggregation of models that together represents the whole system with respect to a set of related concerns. For example, a collection of structural models would (often) form a structural view. We use terms like “operational view” where others have used terms like “operational architecture.” A view belongs to a particular architectural description. A *viewpoint* captures the rules for constructing and analyzing a particular kind of view. It is a template for a view and can be reused across many architecture descriptions. *View type* was considered as an alternative term for viewpoint because of the strong analogy of view and viewpoint to instance and type; but we chose "viewpoint" because of its use in existing standards and the requirements analysis literature. Separating the two concepts is important for writing standards, although not essential if one’s scope is simply the description of a single system.

A fundamental choice in the content of IEEE 1471 is that we allow using organizations to select the viewpoints most appropriate for their use and then require them to write (that is “declare”) those viewpoints—rather than prescribing a set of viewpoints and methods for building views. This greatly reduces the apparent normative content of the document, but avoids otherwise unresolvable problems in establishing a consensus when such a diversity of view methods are currently in use. Perhaps this diversity can be brought into a common framework of viewpoint definitions, but further standardization must await additional research and practical experience.

Because no specific viewpoints are mandatory each implementing organization must decide, based on its own needs, what viewpoints are appropriate for its efforts. This need to decide on what viewpoints to use on a case-by-case basis is built into the requirements of IEEE 1471 because stakeholders must be explicitly chosen for each architecture description, and viewpoints that address those stakeholders' concerns must likewise be chosen for each description.

7. What do viewpoints look like and how do I write them?

The key idea is that a viewpoint identifies a set of concerns about an architecture, and how the architectural description will address these concerns, in terms of languages and notations, models and analytical techniques, methods, etc. In our experience, an organization will develop a library of viewpoints as it gains experience with architecting. At the same time, each system is different, and often an existing viewpoint must be modified or a new viewpoint must be defined for the specific system at hand. The recommended practice provides a number of example definitions of viewpoints in its informative annexes.

8. Does IEEE 1471 require or define a process?

No. It defines what you should have when you claim to have an architectural description, but it does not say how to get one. (It does not even say you must have an AD. Naturally, we think it is a good idea to have an AD for many situations, but IEEE 1471 does not say you have to have one.)

However, using IEEE 1471 within the context of a well-defined process is definitely a good practice. Determining a suitable process is a task for the implementing organizations. While eliminating process elements makes the application of the recommended practice somewhat less obvious, it was essential in finding a set of architectural practices on which community consensus exists. There was much more consensus on the contents of an AD than on the process for constructing an AD.

9. We are already using another architecture framework (e.g., C4ISR, JTA, RM-ODP, etc.), what do we need IEEE 1471 for?

In brief, IEEE 1471 is not a replacement for these other standards, it is an organizing framework. For example, from an IEEE 1471 perspective the requirements of C4ISR can be cast as requiring ADs to have three particular viewpoints selected, the viewpoints corresponding to the three C4ISR views (operational, system, and technical views). Seen this way, C4ISR is an additional requirement placed on ADs for certain classes of system that facilitates the evaluation of those ADs from the perspective of the C4ISR group. One goal of IEEE 1471 is to facilitate the evolution of more domain specific architecture and architecture description standards by installing common language and concepts. IEEE 1471 may be very useful for extending the architecture frameworks that exist and reconciling them with each other. Viewed through the lens of IEEE 1471 some existing architecture frameworks may be better understood. For example, IEEE 1471 forces the question of explicitly considering the stakeholder audience for each architecture description. A standard that provides information required for one group of stakeholders may be a good standard, but IEEE 1471 illustrates its deficiency when additional stakeholders not included in the description standard are introduced.

Annex D of the Recommended Practice addresses its relationship to IEEE/EIA 12207 and ISO RM-ODP. We think that other standards can be similarly recast as additional normative restrictions on IEEE 1471

ADs. If this is true it highlights the value of IEEE 1471. IEEE 1471 becomes an organizing framework for ADs in different domains, and potentially a means for understanding between domains. It also highlights that many existing architecture standards do not specify a set of views large enough to satisfy all concerns, just those of particular stakeholder groups. There is nothing wrong with this, as long as system developers do not think that an incomplete expression of a system's architecture is actually anywhere near complete.

How can I find out more about IEEE 1471?

Visit the IEEE 1471 web site, http://www.pithecanthropus.com/~awg/ for up-to-date information on IEEE 1471, or send mail to ieee-awg@davebert.mitre.org

Copies of IEEE Standard 1471–2000 may be obtained from:

IEEE Standards Office
P.O. Box 1331
Piscataway, NJ 08855-1331

Author Information

Mark W. Maier was educated at Caltech and the University of Southern California. He received B.S. and M.S. degrees from Caltech, and the EEE and Ph.D. in Electrical Engineering from USC. While at USC he was a section head at the Hughes Aircraft Company, Radar Systems Group where he led signal processing algorithm design efforts in advanced radars and electronic warfare systems, and held a Howard Hughes Doctoral Fellowship. He was also a corporate instructor in the Hatley-Pirhai structured analysis and design method. He is currently a Senior Engineering Specialist at the Aerospace Corporation. He has supported a variety of national security space customers including the National Security Space Architect and the National Reconnaissance Office. Previously he was an Associate Professor of Electrical Engineering at the University of Alabama in Huntsville. While at UAH he was the faculty advisor to the SEDSAT-1 student satellite project which successfully launched a microsatellite in 1998.

Maier is the co-author of "The Art of Systems Architecting" with Eberhardt Rechtin. He was a member of the IEEE 1471 Recommended Practice for Architectural Description writing team and was the chair of the ballot resolution committee.

Mark W. Maier
The Aerospace Corporation
Suite 600
15049 Conference Center Dr.
Chantilly, VA 20151
703-633-5350
mark.w.maier@aero.org

David Emery is a Principal Engineer in MITRE's Army Information Systems department, providing systems and software engineering on a variety of military command and control and weapon systems. He previously worked for Hughes Aircraft of Canada, Siemens Research and Computer Sciences Corporation, and served on active duty with the U.S. Army.

Mr. Emery received his B. S. in Mathematics from Norwich University, Northfield, VT in 1978. He was commissioned a Second Lieutenant, Field Artillery, and served in a variety of artillery and automation assignments on active duty. He became interested in Ada and large-scale software engineering problems while in the military, and his professional career has been involved in Ada, software engineering and software standardization.

He is active in both the IEEE and the ACM, and has participated in several international standards activities. His IEEE activities include Technical Editor of IEEE P1003.5, the Ada Binding to POSIX and contibuted to the recently approved IEEE Std 1471, Recommended Practice for Architecture Descriptions for Software Intensive Systems. He has served as Secretary and Treasurer for ACM's Special Interest Group on Ada, and as a member of ACM's Technical Standards Committee. Within ISO, he has been a member of the US Delegation to ISO/IEC SC22 (Programming Languages and Interfaces) and to ISO/IEC SC22 WG9 (Ada), and has chaired WG9's Ada Uniformity Rapporteur Group.

Mr. Emery has been honored with the IEEE Third Millenium Medal, Outstanding Contribution and Meritorious Service awards, and selection to the IEEE Computer Society's "Golden Core". SIGAda recently awarded him its Outstanding Contribution Award. He is published on Ada programming language bindings, software portability and architectural approaches for software-intensive systems. His paper *Experiences Applying a Practical Architectural Method* won Best Paper award at Ada-Europe '96.

David Emery
The Mitre Corporation
703-883-7606
emery@mitre.org

Rich Hilliard is Chief Technology Officer and co-founder of ConsentCache, Inc. Previously, he was Director of Architecture at ephibian.com (formerly, ISIS 2000) where he led their architecture consulting efforts. From 1994-1998, he was a lead architect at the MITRE Corporation, where he was a founding member of the Chief Architects' Office of the Air Force's Electronic Systems Center (ESC/DIA). In that capacity, he was a major contributor to the architecture for the Air Force's Integrated Command and Control System (C2STA). From 1991 to 1994 he was a member of the Ada 9X Mapping/Revision Team, which developed the ISO standard for Ada 95. In the early 1980s, he was one of the developers of the IDEF methods for the Air Force's Integrated Computer-Aided Manufacturing (ICAM) program.

He served as technical secretary of the IEEE Architecture Working Group which has recently completed IEEE-Std-1471-2000, Recommended Practice for Architectural Description of Software-Intensive Systems.

Rich Hilliard
ConsentCache
Rh@consentcache.com