

Understanding Architectural Perspectives*

Rich Hilliard
rh@wn.net

March 7, 2005

Abstract

This note analyzes Woods' and Rozanski's notion of *architectural perspective* in light of the conceptual framework of IEEE Std 1471.

1 Introduction

In their forthcoming book [1], Nick Rozanski and Eoin Woods introduce the notion of an *architectural perspective* as a type of architectural knowledge, applying it to the construction of architectural descriptions (ADs), as those are defined in IEEE Std 1471 [2]. The authors, with Wolfgang Emmerich, have also written a paper [3] (henceforth: **WER**) explaining this notion.

In this note, I analyze the notion of architectural perspective in light of the conceptual framework of IEEE 1471, based upon the material in **WER**. The reason for my analysis is two-fold: (1) I'm interested in understanding kinds of architectural knowledge; and (2) the 5-year revision of IEEE 1471 is now underway. One important concern for that revision is the consideration of new constructs within the practise of software architecture which should be reflected in the standard. A related concern for the revision is to clarify anything in the previous version which was not sufficiently well-understood to be put into practise.

2 Specific Comments

Managing Quality. **WER**'s Introduction is an eloquent rationale for why architects must manage quality-related concerns, or "quality properties" of the systems they architect.

We call our approach *Architectural Perspectives* and it provides a framework for structuring knowledge about how to design systems to achieve particular quality properties. In many ways, Perspectives are similar to Architectural Viewpoints [10], but whereas a particular Viewpoint conventionally advises on how to create and describe a particular type of architectural structure [10, 14, 9, 18], a Perspective relates to the cross view concerns of a particular architectural *quality property*.

As the authors note, their purpose for introducing architectural perspectives is similiar – in fact identical to – IEEE 1471's purpose for architectural viewpoints: as mechanisms for capturing a useful kind of architectural knowledge. However, the second part of the quotation suggests an interpretation of IEEE 1471 unintended by the standard's authors: *nothing in IEEE 1471 restricts architectural viewpoints to being about*

* A future version of this note may take on viewtypes, aspects and textures in the context of the IEEE 1471 conceptual framework.

structure. In fact, the notion of structure is intentionally left undefined in IEEE 1471, and plays no role in any of its stated requirements. There was an extended debate about this during the creation of IEEE 1471 (see **Sermon**, below).

The other issue here is that of a perspective “relat[ing] to the cross view concerns of a particular architectural quality property,” I return to this issue below.

Minor historical note.

Finkelstein *et al.*, who have coined the term viewpoints, suggested them as a conceptual framework to define templates whose instances describe software systems from different perspectives [8, 17].

Actually, the usage of the term viewpoint in Software Engineering is *much older* than [4]. The first usage I’m able to track down is by Doug Ross, inventor of Structured Analysis, in the 1970s [5, 6]. The usage of viewpoint in IEEE 1471 is intentionally patterned after his work. For a discussion of subsequent usage of the term, including Finkelstein *et al.*’s in Requirements Engineering, see [7].

Defining perspective.

Our definition of Architectural Perspective is a collection of activities, checklists, tactics and guidelines to guide the process of ensuring that a system exhibits a particular set of closely related quality properties that require consideration across a number of the system’s architectural views. In other words, a Perspective is a collection of guidance on achieving a particular quality property in a system.

This is consistent with the permissible content of a viewpoint, which may contain: “Heuristics, patterns, or other guidelines to assist in synthesis of an associated view” [IEEE 1471, clause 5.3].

Extensible framework.

We developed the concept of the Architectural Perspective (or just “Perspective”) in order to provide an extensible framework, within which we could capture knowledge about designing systems that need to exhibit specific quality properties.

This is also analogous to the intent of introducing architectural viewpoints in IEEE 1471. The authors of the standard chose not to prescribe a fixed set of required architectural viewpoints as a part of an AD; introducing, instead, a mechanism by which users of the standard could define viewpoints in a consistent manner:

“Because of the wide range of opinion on selecting appropriate views, this recommended practice does not prescribe a fixed set of views, rather, it introduces the concept of a viewpoint to designate the means used to construct individual views.” [IEEE 1471, Annex B]

Information Content of a Perspective vs. a Viewpoint.

A Perspective contains the following information:

- the Concerns that the perspective is addressing;
- the Applicability of the perspective to the different possible architectural views of a system (and the types of system to which the advice within it relates, if this is not obvious);

- a set of possible Activities that are suggested as part of the process of achieving the quality property (ideally related to each other via a process to follow);
- a set of proven Architectural Tactics (i.e. design strategies) [4] that the architect can consider as part of their design;
- a list of common Problems and Pitfalls that the architect should be aware of and common solutions to them; and finally
- a Checklist that the architect can use to ensure that nothing has been forgotten.

The first bullet refers to concerns (plural). At other places in **WER**, the point is made that a perspective pertains to a *single* quality concern. This needs to be clarified. For the remainder of this note, I will assume that a perspective addresses a single architectural concern. In IEEE 1471, a viewpoint may address multiple concerns; this seems to be the right generalisation.

All of the above items readily map to the information content of a viewpoint, with the exception of the second bullet. Bullet two splits into two cases, which should be handled separately. The first part of bullet two pertains to the architectural views to which the perspective applies. This is different from a viewpoint, because in IEEE 1471, a viewpoint applies to *exactly and only one view*. This situation, I return to below. The other case referred to in this bullet is the applicability of the perspective to “types of systems”. This can be dealt with within the existing definition of viewpoint via heuristics or guidelines associated with the viewpoint.

Perspectives and architectural models.

This said, applying many perspectives can actually produce outputs such as threat models, performance models and so on, but these are really supporting information rather than first-class architectural design artefacts.)

There are two issues here: (ι_1), What constitutes “first-class architectural design artefacts” – i.e., what is a proper part of an AD (architectural description)? And (ι_2), Where does that information reside within an AD? IEEE 1471 has nothing to say about question ι_1 : it leaves that to users of the standard as a method/process issue. The second question is more interesting from the IEEE 1471 point of view, because of the following consequence: are there artifacts (or models) which are not a part of any architectural view in the AD? If the model addresses a stated architectural concern, and that model is not associated with at least one view, that breaks the IEEE 1471 requirement that all concerns be addressed by at least one view.

Single Quality Property.

Each Perspective addresses a single quality property, which means that for any complex system the architect has to apply a number of them and there is no guarantee that the advice in each will be compatible. Indeed, you would expect the advice in a number of them to conflict (between performance and flexibility concerns in different Perspectives for example) and the architect needs to resolve these conflicts when they arise.

There are two issues here. First, is that a perspective addresses exactly one quality property. Second, is that it is hard to satisfy multiple concerns without conflicts. IEEE 1471 has nothing to say about the latter issue. On the first issue, a viewpoint, unlike a perspective, apparently, is not restricted to addressing a single quality-related concern.

Redundancy in Views.

Our experience suggests that creating quality property based viewpoints is not an effective approach, and in particular, when we have tried to create quality property specific views of a system, we have found that a significant problem is the amount of redundancy introduced into the architectural model – redundancy that often results in the architectural description being abandoned as it is too hard to maintain.

It is hard to address this qualitative issue in a definitive fashion, but in my experience, redundancy of this kind can be reduced in three ways:

- refactoring of the viewpoints;
- use of architectural models [IEEE 1471, clause 5.4] to share, rather than repeat, details among views;
- careful use of abstraction in the definition of the entities which comprise each viewpoint. (Slogan: *Render unto Caesar that which is Caesar's.*) E.g., a security viewpoint which introduces three kinds of entities: Subject, Principals, and Mechanisms. Elements of other views are (exhaustively) categorized as subjects and principals. The Security viewpoint defines rules governing which mechanisms apply to which categories. The remaining details of the entities reside in other views.

3 Analysis

An architectural perspective plays the same role as an architectural viewpoint, except that: (Δ_1) it may apply to more than one view, whereas a viewpoint applies to exactly one, and (Δ_2), a perspective addresses exactly one quality-related concern (or quality property), whereas a viewpoint may address one or more concerns. The other differences suggested by **WER** have been shown in the previous section to be non-issues.

Assuming that the notion of architectural perspective is useful, can it be used in combination with IEEE 1471, given these differences? There are two options: (a) apply perspectives as a part of an architecture method, outside the scope of IEEE 1471, but leading to IEEE 1471 conforming ADs; or (b) assimilate the notion of perspective into the IEEE 1471 conceptual framework, adjusting the IEEE 1471 requirements or the definition of perspective, or both, to accommodate.

Under (a), (Δ_1) and (Δ_2) do not directly arise as issues. However, under (a) the problem with perspectives is that, as defined, their use may compromise the ability of a resulting AD to conform to IEEE 1471. Consider the requirement:

Each stakeholder and each concern identified in an AD in accordance with 5.2 shall be addressed by at least one viewpoint selected in accordance with 5.3. [IEEE 1471, clause 5.3].

The concerns allocated to a perspective may not be allocated to a viewpoint, therefore there will be architectural concerns not addressed by some viewpoint, therefore the resulting AD will not satisfy the above requirement. The alternative would be that perspectives only address concerns which are always already allocated to some architectural viewpoint. This seems like a strange situation.

Under (b), consider two ways of adding architectural perspectives to the IEEE 1471 conceptual framework: (i) creating a new construct, parallel to viewpoints, or (ii) “subclassing” viewpoints to make perspectives.

Introducing perspectives in parallel with viewpoints, would result in two very similar concepts with different rules, or in the relaxation of the existing rules on viewpoints to encompass perspectives, or changes to perspectives as defined.

In the interests of simplicity, or parsimony, one concept is always to be preferred over two, all other things being equal. Introducing perspectives in parallel with viewpoints is of questionable utility. Relaxation of the rules for viewpoints to incorporate perspectives, is not warranted for most viewpoints. It also might lead to very lazy architectural practices, where everything is thrown into a single “view”; that would negate the well-understood values of separation of concerns, etc. I return the possible changes to the definition of perspectives at the end of this section.

Introducing perspectives as a subclass of viewpoints, one would define an architectural perspective as *just like* an architectural viewpoint except for (Δ_1) and (Δ_2) .

By making perspective a subclass of viewpoint, one can avoid the problem above with (a) pertaining to coverage of all concerns under some viewpoint, since a perspective is a viewpoint.

So the discussion reduces to the issues pertaining to (Δ_1) and (Δ_2) . Can these two issues above be resolved?

On (Δ_1) : The IEEE 1471 rule, *one viewpoint – one view*, is a strong tenet of IEEE 1471. The slogan motivating this rule was:

view : viewpoint :: object : class.

The authors were inspired in part by the extreme complexity of things like C++’s rules for multiple inheritance to avoid such situations – even though the burden of complexity in this case would fall not on the text of the standard, but on users of the standard.

Another operative slogan for the authors of IEEE 1471 was:

viewpoint : view :: legend : map

A view is a map; the IEEE 1471 authors wanted readers to be able to look in one place (i.e., the viewpoint) for the legend associated with understanding that map (the view). Introducing perspectives, as defined, would result in views having multiple legends: at least one viewpoint and possibly one or more additional perspectives which a reader would have to look at to understand a particular view.

On (Δ_2) : A perspective is defined to address exactly one quality concern. This seems like an unnecessarily harsh restriction, Is anything lost by following the IEEE 1471 model of “allocating” one or more architectural concerns to a viewpoint/perspective? This generality seems especially compelling given that there is no strict definition establishing the “granularity” of what constitutes an *architectural concern*.

4 Conclusion

I have analyzed **WER**’s notion of architectural perspective.

The original reason for introducing perspectives was to address qualities, but this does not distinguish the notion from the original intent of viewpoints. A viewpoint is intended to address one or more architectural concerns, which may include one or more qualities. In IEEE 1471, there is no stipulation that these concerns be limited to “structural” concerns, as suggested in the proposal for perspectives – in fact the notion of structure does not arise in IEEE 1471.

Therefore, the notion of architectural perspectives aligns with the existing notion of architectural viewpoint, but for two differences. These differences were analyzed in the previous section. It seems arbitrary to restrict a perspective to a single concern.

This leaves the issue of the application of perspectives to multiple views. As we begin the IEEE 1471 revision, it is worth considering specific cases, as to whether this generalisation is warranted, or can be adequately handled with the existing mechanisms.

My personal view, is that reduced to this single issue, introducing perspectives into IEEE 1471 is not warranted without further evidence, but others may see it otherwise. It is worth consideration.

References

- [1] Nick Rozanski and Eoin Woods. *Software Systems Architecture: Viewpoint Oriented System Development*. Addison Wesley, 2005.
- [2] *ANSI/IEEE Std 1471–2000 Recommended Practice for Architectural Description of Software-Intensive Systems*, October 2000.
- [3] Eoin Woods, Wolfgang Emmerich, and Nick Rozanski. Using architectural perspectives. Unpublished draft, dated August 2004.
- [4] A. Finkelstein, J. Kramer, B. Nuseibeh, L. Finkelstein, and M. Goedicke. Viewpoints: a framework for integrating multiple perspectives in system development. *International Journal of Software Engineering and Knowledge Engineering*, 2(1):31–57, March 1992.
- [5] Douglas T. Ross. Structured Analysis (SA): a language for communicating ideas. *IEEE Transactions on Software Engineering*, SE-3(1), January 1977. Also appears in *Programming methodology : a collection of articles by members of IFIP WG2.3* edited by David Gries. New York : Springer-Verlag, 1978.
- [6] Douglas T. Ross and Kenneth E. Schoman. Structured Analysis for requirements definition. *IEEE Transactions on Software Engineering*, SE-3(1), January 1977.
- [7] Rich Hilliard. Views and viewpoints in software systems architecture. Position paper from the *First Working IFIP Conference on Software Architecture*, San Antonio, 1999.

A A Brief Sermon on Structure

In the beginning, architecture was without form. Then components and connectors spread across the landscape ...

The work to define IEEE 1471 began in 1995. The IEEE Architecture Working Group was a mix of industry and academic individuals, from both the software engineering and systems engineering communities.

Inspired by the work of Barry Boehm *et al.* on stakeholders and concerns, multi-disciplinary approaches to systems engineering (e.g., Maier and Rechtin), and the “existence proof” established by Philippe Kruchten’s 4+1 view model architecture methods, the authors of IEEE 1471 adopted an approach to software architecture based on multiple views, stakeholders and concerns.

It was at roughly the same time that “Software Architecture” became a popular phrase in the academic literature. The IEEE 1471 conception was regarded as somewhat counter to a prevalent, then-current view of software architecture along the lines of the slogan:

The software architecture of a system is the large-scale organization of the system in terms of its components and connectors.

The idea of this approach was that the architectural structure embodied in the components and connectors was primary, and that other concerns were addressed via annotations of that primary structural view.

This was a loud and interesting debate. In the end, the IEEE 1471 authors stuck with their original conception that architecture was inherently concern-oriented and multi-viewpointed, and that structure was an important, but not a distinguished, view of the system.

In retrospect, this was the right decision. I suspect that from a rational vantage point, *architectural structure* was a pre-theoretical notion, about as useful in software architecture as epi-cycles are in the Copernican theory of the solar system. What we have learned since 1995, is that “structure” is only what we can attain by defining our viewpoints and the rules for their construction.