

Aspects in Architectural Description

Report on a first workshop at AOSD 2007

Paul Clements
SEI
clements@sei.cmu.edu

David Emery
DSCI
demery@dsci-usa.com

Rich Hilliard
r.hilliard@computer.org

Philippe Kruchten
UBC
pbk@ece.ubc.ca

Abstract

The First Workshop on Aspects in Architectural Description was held in Vancouver, British Columbia as a part of Aspect-Oriented Software Development (AOSD) 2007. This workshop report presents a summary of the topics addressed, contributions and issues raised, and workshop conclusions.

Keywords: aspects, architectural concerns, architectural description, architectural models, architectural views and viewpoints

Introduction

The original definition of an *aspect* by Gregor Kiczales is that of a “well-modularized cross-cutting concern.” Aspects have made a visible impact on software design and programming. This is because the concept contributes to our understanding of system construction, modularization, and the articulation of important functional and non-functional concerns.

The theme of the First Workshop on Aspects in Architectural Description was, “How do aspects influence the description of architectures?” This is of interest to both software architects and system architects. The on-going revision of IEEE 1471 (now also international standard ISO/IEC 42010), *Recommended practice for architectural description of software-intensive systems*, provides an opportunity to influence thinking in architecture, particularly with respect to how aspect concepts interact with the use of architectural viewpoints and the ability of aspects to capture non-functional concerns about architectures within architectural descriptions.

As demonstrated in the *Early Aspects* series of workshops [1], aspects are applicable not only in programming but to the earlier stages of software systems development including requirements definition and design and their use for specifying architectural properties: both emergent properties of the system as a whole and the contributions of architectural components to emergent properties, including the ability to check architectural components against aspect “requirements” for that component.

It was a goal of this workshop to offer insights, guidance, and examples for the community using aspects and serve as useful input into the revision of ISO/IEC 42010 to improve support for aspect-oriented approaches to architectural description.

Topics of Interest

The Call for Participation raised these questions of interest to the workshop:

- How do aspects influence the description of architectures? How do we specify aspects within an architectural description?
- How do aspects relate to other practices of architectural description? Currently, architectural descriptions are often orga-

nized into views each addressing one or more concerns. How do aspects interact with this prevailing approach?

- Can aspects be used within views, between views?
- How do aspects contribute to, and participate in, viewpoint definitions?
- How do aspects relate to views and viewpoints, and to other architectural description mechanisms such as viewtypes [3], textures [7] and perspectives [8]?
- Are aspects useful in systems architecture as well as in software?
- How are aspects and architectural decisions [6] related? Can aspects be used to record certain types of architectural decisions? Can aspects be used for “model checking” or “contract checking” of components within an architecture to ensure the component meets the intent of the architecture?
- What forms of aspect advice (i.e. operators) make sense with architectural aspects?
- In an architectural description constructed with multiple viewpoints, what is the base language for architectural aspects? Is the base a particular viewpoint language (e.g., components-and-connectors), any viewpoint language, all viewpoints?

Participants

Attending the workshop were:

Agung Fatwanto (UBC), Alessandro Garcia (Lancaster), Anthony Finkelstein (ICL), Brian Barry (Bedarra), Claudio Sant’Anna (PUC Rio), David Emery (DSCI), Davide Falessi (UBC), Elisa Yumi Nakagawa (USP), Jon Oldevik (U Oslo), Nelis Boucké (KULeuven), Paul Clements (SEI), Philippe Kruchten (UBC), Rich Hilliard, Sebastian Herold (Clausthal), Steven Op de Beeck (KU-Leuven), and Urjaswala Vora (CDAC).

Summary

The workshop was held for a full day, on 12 March 2007. Ten position papers were accepted, and six of these were presented during the workshop. One of the organizers also provided a paper. All of these submissions are listed in Table 1. All workshop materials are available from the workshop web site [2].

- "The Symbiosis between View and Aspect," Eduardo Barra, Anabel Fraga and Juan Llorens.
- "View Composition as a First-Class Concept in Architectural Descriptions," Nelis Boucké, Danny Weyns and Tom Holvoet.
- "Are Architectural Aspects Style-Dependent?" Christina Chavez, Alessandro Garcia and Thais Batista.
- "Aspect-Oriented Concepts and Architectural Views," Sebastian Herold.
- "Representing Aspect-Based Architecture of Software Engineering Environments," Elisa Yumi Nakagawa and José Carlos Maldonado.
- "Aspectual vs. Component-based Decomposition: A Quantitative Study," Steven Op de beeck, Dimitri Van Landuyt, Johan Gregoire, Riccardo Scandariato, Wouter Joosen, Andrew Jackson and Siobhán Clarke.
- "Architectural Aspects in UML," Jon Oldevik and Øystein Haugen.
- "On the Modularity Assessment of Software Architectures: Do my architectural concerns count?" Cláudio Sant'Anna, Eduardo Figueiredo, Alessandro Garcia and Carlos J. P. Lucena.
- "Aspects in Architectural Description of Evolving Systems," Urjaswala Vora.
- "Aspects in AD: Thoughts about Quality of Service," David Emery.
Presentation on [linkit](#), Anthony Finkelstein.

Table 1 — Summary of Submissions

Themes

Several themes arose during the workshop's deliberations: relation of aspects to IEEE 1471; documentation of aspects; aspects versus views. Issues raised by the presentations and discussions included:

- the need for a working definition of *architectural aspects* (see working definition below);
- the need for good examples of architectural aspects (not just obvious ones from aspect-oriented programming). There are useful examples in the submitted papers, and there was discussion of working with the HeathWatcher case study being studied by the *Early Aspects* community.
- Aspects and views—should they be the same or different? Each can be considered a kind of module of architectural descriptions whereas views (at least in IEEE 1471) are subject to a wholeness criterion: each view must represent the *whole* system with respect to some set of concerns.
- Are aspects useful within views or between views? R. Hilliard suggested both kinds are found, and correlated these situations with A. Ran's textures [7], and Rozanski & Woods' perspectives [8], respectively. Architectural textures capture the "recurring microstructure" within a view. Aspects and architectural perspectives both crosscut views.
- Should aspects be first-class entities in IEEE 1471 and how?
- Can aspect oriented design bring more to conceptual framework than composition? What's different from view?

- Do aspects pertain to view composition, or model composition, or both? It was recognized the architectural aspects are intimately linked to the definitions of architectural models and architectural views (see Figure 1, taken from IEEE 1471). N. Boucké showed how to do view composition via model relations, which would require more rigorous definition of model in IEEE 1471.

The following topics were identified, but not addressed during the workshop:

- What kinds of composition (or weaving) mechanisms are needed, or useful for architectural aspects?
- Are architectural aspects (a)symmetric, or both?
- How do aspects relate to the expression of architectural decisions and their rationales?
- Verification of aspects.

Taxonomic Attributes of Architectural Aspects

The workshop identified some attributes to classify the contributions in the submissions and deliberations, and other work.

- P. Clements proposed the following 3-way classification as follows:
 - 1) *Inherited* (coming from requirements or other stakeholders);
 - 2) *Created* (introduced by the architect for the architecture's sake); and
 - 3) *Prescribed* (by the architect downstream to developers)

- What are the entities which are being crosscut by the aspect? Architectural aspects were found that crosscut views, components and connectors.
- What is the base language? What are the base language unit(s)? The two most prevalent languages represented in the

papers were components and connectors-based ADLs and UML.

- What concerns are being addressed by the aspect?

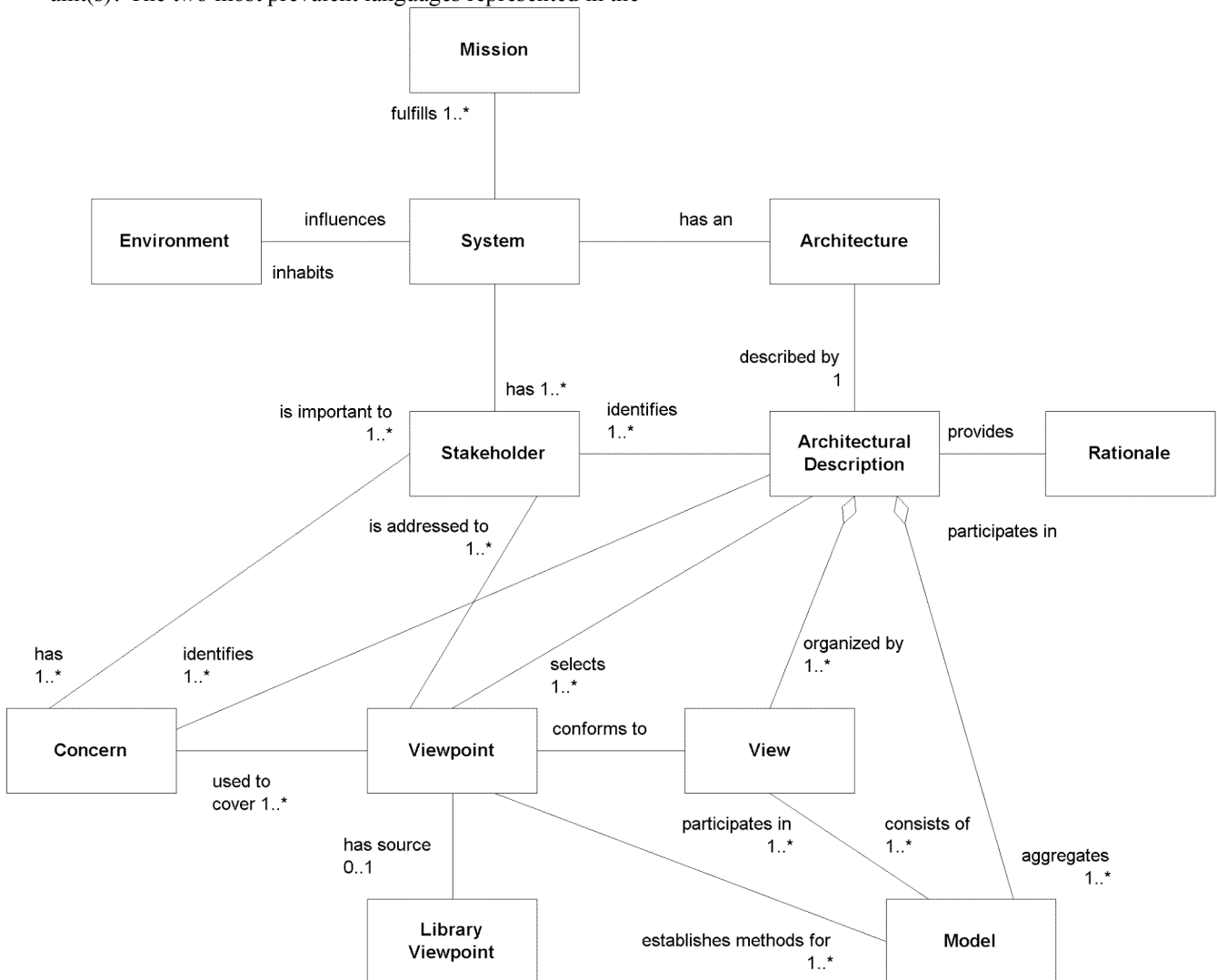


Figure 1 — The IEEE 1471:2000 Conceptual framework (or “metamodel”)

Conclusions

In wrapping up the workshop, the participants tried to formulate a working definition of architectural aspect; and identified potential benefits and weakness of an aspect-oriented approach to architecture. These are documented below.

Working Definition. An *architectural aspect* is an encapsulated artifact (e.g., module, document) addressing a cross-cutting concern, that is, a concern (from the problem domain) cutting across one or more of the elements in one or more of the many decompositions in other architectural artifacts (e.g., views, perspectives, models, or other aspects).

Benefits

- Greater conceptual integrity
- Management of multiple views
- Increased productivity: reuse of architectural aspects
- More effective communication: upstream (with requirements) and downstream (with design, testing, and integration)
- Appropriate level of consistency: savings at the implementation level; Even in the face of evolution
- Provide tools for synthesis and analysis

Potential Drawbacks

- Increased complexity
- Obfuscation of the architectural description
- Fragmentation of the architectural description
- Misuse and abuse of a concept suited for programming

Future, Next Steps

The participants entertained ideas for future steps on aspects in architectural description. These were:

- to feed the contributions and results of the workshop to ISO/IEC/SC 7/WG 42 as inputs to the IEEE 1471 | ISO/IEC 42010 joint revision;
- to present the results to IFIP WG 2.9 Requirements Engineering and IFIP WG 2.10, Software Architecture;
- to identify architectural aspects in practice; and
- to propose a follow-on workshop. Possible venues named were: FSE, EWSA, WICSA, and AOSD.

Acknowledgements

The organizers wish to thank all of the workshop participants for their contributions and the AOSD 2007 organizers who provided a comfortable infrastructure for the workshop, including the first sunny day in quite a while.

References

- [1] Early Aspects Initiative, web site: <http://early-aspects.net/>.
- [2] First Workshop on Aspects in Architectural Description, web site: <http://aosd.net/workshops/aarch/2007/>.
- [3] Clements, P., et al. (2002). *Documenting Software Architectures: Views and Beyond*, Boston: Addison-Wesley.
- [4] Hilliard, R. (1999). Aspects, Concerns, Views, ... *Proceedings of the OOPSLA'99 Workshop on Multi-Dimensional Separation of Concerns*.
- [5] IEEE Std 1471:2000 – *Recommended practice for architectural description of software-intensive systems*. Los Alamitos, CA: IEEE.
- [6] Kruchten, P., Lago, P., and van Vliet, H. (2006). Building up and reasoning about architectural knowledge. In C. Hofmeister (ed.), *Proceedings of QoSA 2006* (LNCS Vol. 4214, pp. 43–58). Berlin: Springer-Verlag.
- [7] Ran, A., (2000). ARES Conceptual Framework for Software Architecture. In Jazayeri, M., Ran, A., van der Linden, F. (eds.), *Software Architecture for Product Families Principles and Practice* (pp. 1–29). Boston: Addison-Wesley.
- [8] Rozanski, N., and Woods, E. (2005). *Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives*. Boston: Addison-Wesley.