

# Aspect-Oriented Concepts and Architectural Views

## Position Paper

Sebastian Herold

Clausthal University of Technology  
Institute for Computer Science - Software Systems Engineering  
Julius-Albert-Straße 4, 38678 Clausthal-Zellerfeld, Germany  
sebastian.herold@tu-clausthal.de

### Abstract

There are crosscutting concerns in architectural description as well as in other fields of software engineering. Views are an appropriate and established instrument to deal with different concerns regarding software systems architecture and its description. On the other hand, aspect-oriented concepts are increasingly influencing software architecture. This paper is going to discuss some issues regarding aspect-oriented concepts and views that were brought up by describing a simple example.

### 1. Introduction

Architecture descriptions are often structured by views to encapsulate the concerns a certain group of stakeholders may have regarding the architecture [3, 7]. Thus, as [6] works out, views can be seen as modules on the architecture description level. However, it has been shown that crosscutting concerns cannot be easily modularized with one-dimensional paradigms [10]. To tackle these problems, methods from aspect-oriented programming could be helpful in architecture description and specification to integrate views.

This raises questions about the relation between views and aspects and how to describe them. Can crosscutting architectural concerns modelled by a view which crosscuts other views and, therefore, can be called an architectural aspect? How does weaving relate to other composition techniques? If certain views have

characteristics of aspects, what is the role of viewpoints? Can arbitrary views be woven, or does weaving take place between partial views inside a superior or dominant view?

In the following, we would like to stimulate further discussion upon those topics by introducing a simple example and discussing some problems.

### 2. Example

The example system we chose is a three layer architecture information system [1] with rich client distribution [9]. The control layer is integrated into the presentation layer which is distributed to the client environment at runtime. Application layer and data layer are distributed to the server and communicating via some remote communication framework. Figure 1 shows a non-modularized and coarse-grained description of the system's architecture.

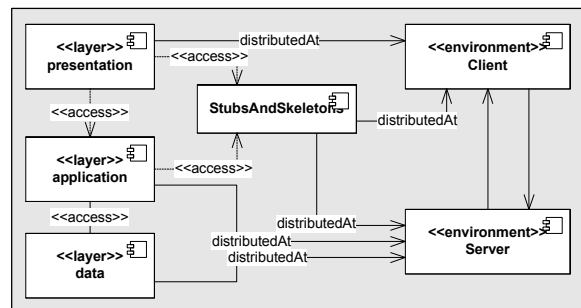
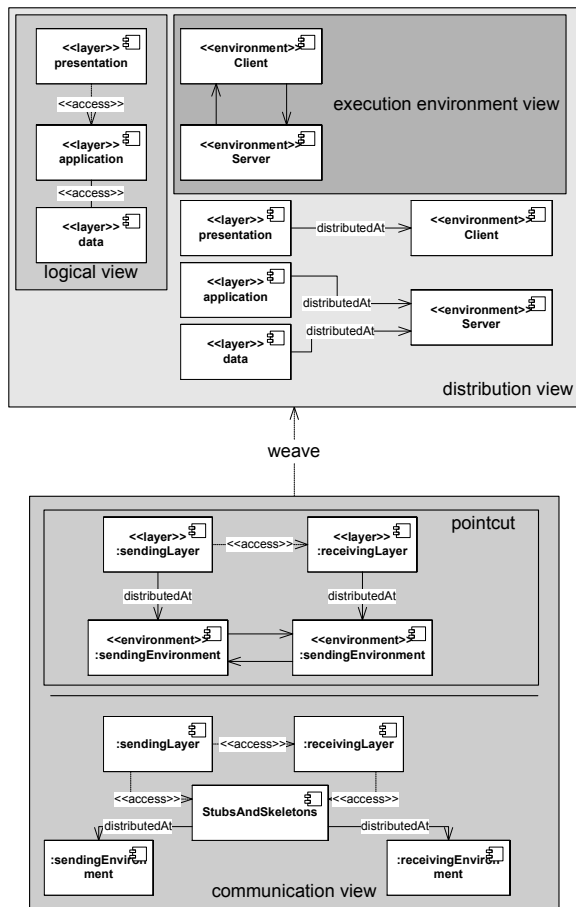


Figure 1: Coarse-Grained Description of Example



**Figure 2: Views of Example System**

There are different views the software architect (and others) would be interested in. The logical view would show the available layers and the dependencies between them. The execution environment provides the information that the system is distributed to different environments, namely a client and server. Additionally, the software architect would look at the communication framework separately.

If we would like to specify the system's architecture by strictly separating these views, the architecture description could look somewhat like in Figure 2 which shows the views in some pseudo-notation. Logical layers are explicitly modeled by the logical view, as well as client and server in the execution environment view. The distribution view is built by explicitly stating

how layers are distributed to client and server environments.

The lower part of Figure 2 shows another possibility how views can be integrated. It shows the communication view which describes which mechanisms are used to communicate between different execution environments. It defines a pointcut to describe where this view has to be woven into the distribution view. It states that those points of the distribution view are selected where one layer accesses another that is distributed at a different environment. The lower part describes the strongly simplified weaving which adds access relations between the layers affected by the pointcut and some stubs and skeletons module of the communication view. Of course, in reality the weaving would be much more complex.

Although this example is a rather coarse-grained architectural description, it is realistic with regard to the views introduced. In the following, we will discuss some issues concerning the description and aspect-oriented concerns.

### 3. Discussion

Obviously, we have to deal with two different ways to integrate the views of the example. On one hand, we have an explicit binding between logical and environment view to form the distribution view. Both views can be described without references to each other – the structural concatenation is described in the distribution view itself.

In contrast, the communication view is not integrated by explicitly naming the connection point but by defining a pattern of the distribution view where it can be woven into. Although this “pointcut” selects only one point in the example system, the concern is potentially crosscutting in distributed systems.

Additionally, it seems that the logical view plays a dominant role in constructing the system above. The weaving as well as the composing of the distribution view affects the logical view. If you think of views to software systems which, similar to our communication view, deal with technical crosscutting, the logical architecture is

always involved and crosscut. This indicates that the logical architectural view could be treated as core concern in an asymmetrical approach (see, for example, [4]).

However, the question raises whether the communication view as such is reasonable related in the description above. Since it deals with a crosscutting concern of logical components, namely the ability to access or be accessed remotely, it could be interpreted as sub-view of the logical view in the sense that it is completely contained in it which, in general, would lead to some hierarchy between views. If we divide the logical view into functional and crosscutting technical views the communication view could be a single technical view which is woven into the business views (and into other technical views as well) which, of course, could be structured symmetrical [2]. Of course, the communication view still has to reference the higher-level distribution view containing the logical view.

Another question is, which impact such observations have on viewpoints and their relation to views [5]. If viewpoints relate to views like types to instances it would be possible to define whether it is in general crosscutting and which other viewpoint's instances are crosscut.

#### 4. Conclusion

As far as we have seen and discussed, we can point out some issues regarding aspect-oriented concepts in architectural description. If we regard views as modules of architectural description, we can regard views which deal with crosscutting concerns as architectural aspects.

After describing the example above we think that aspect-oriented concepts, basically encapsulating crosscutting concerns, can be realised by views and that weaving is one way of integrating views. This is done by defining patterns of structures inside other views where the crosscutting view is interwoven. Thus, the notational extension of architecture description for aspect-oriented concepts should be not to complex. Maybe, it's even less confusing to speak of views instead of introducing the term "aspect" to the domain of architectural description.

More problematic in our eyes is how to structure views for defining where weaving can be applied. This directly affects the definition of viewpoints. Since we have dependencies between the crosscutting view and the views affected by its pointcut pattern, those dependencies allowed could be defined by the viewpoint to introduce some "type safety".

#### References

- [1] Frank Buschmann et al. *Pattern-Oriented Software Architecture. A System of Patterns*. John Wiley & Sons, 1996.
- [2] Siobhán Clarke, Elisa Baniassad. *Aspect-Oriented Analysis and Design. The Theme Approach*. Addison Wesley, 2005.
- [3] Paul Clements et al. *Documenting Software Architectures: Views and Beyond*. Addison-Wesley, 2002.
- [4] Robert E. Filman et al. *Aspect-Oriented Software Development*, Addison-Wesley, 2005.
- [5] Rich Hilliard. Aspects, Concerns, Views, ... *Proceedings of the OOPSLA'99 Workshop on Multi-Dimensional Separation of Concerns*, 1999.
- [6] Rich Hilliard. Views as modules. Position paper for the Fourth International Software Architecture Workshop (ISAW-4), 2000.
- [7] IEEE Std 1471:2000—*Recommended practice for architectural description of software-intensive systems*. Los Alamitos, CA: IEEE.
- [8] Mary Shaw, David Garlan. *Software Architecture. Perspectives on an Emerging Discipline*. Prentice Hall, 1996.
- [9] Andrew Tanenbaum, Marten van Steen. *Distributed Systems. Principles and Paradigms*. Prentice Hall, 2003.
- [10] P. Tarr, H. Ossher, W. Harrison, S.M. Sutton, jr. N degrees of separation: Multi-dimensional separation of concerns. *In Proceedings of the 21st International Conference on Software Engineering (ICSE '99)*, 1999.