

University of Groningen

Architectural Patterns in Practice

Avgeriou, Paraskevas; Zdun, Uwe

Published in:
EPRINTS-BOOK-TITLE

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2005

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Avgeriou, P., & Zdun, U. (2005). Architectural Patterns in Practice. In EPRINTS-BOOK-TITLE University of Groningen, Johann Bernoulli Institute for Mathematics and Computer Science.

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Focus Group Report: Architectural Patterns in Practice

Paris Avgeriou

*Department of Mathematics and Computing Science
University of Groningen
Groningen, the Netherlands*
paris@cs.rug.nl

Uwe Zdun

*Department of Information Systems
Vienna University of Economics and BA
Vienna, Austria*
zdun@acm.org

1 Introduction

Architectural patterns were one of the very few points, where consensus was achieved in the field of software architecture: their significance is well-established and they are essential to an architecture description. Architectural patterns are widely accepted as recurring solutions that solve problems at the architectural design level, and provide a common vocabulary in order to facilitate communication. Architectural patterns also provide the means to reason for the quality attributes of a software system and help to document the design decisions taken by the architect.

Regrettably, describing, finding, and applying architectural patterns in practice still remains largely ad-hoc and idiosyncratic. Those involved in software architecting often face a number of questions that raise strong debates and cause further problems:

- Are architectural patterns nothing more than raw design solutions? A big part of the literature treats architectural patterns (also known as architectural styles) as design templates without having a context, a specific problem to solve or even a rationale. They may not even be considered generic and “timeless” in the Alexandrian sense, but become much more concrete and focused.
- What is the granularity of architectural patterns? Can some GOF patterns be considered architectural?
- How are all the patterns from the different catalogs related and how can they be combined (e.g. POSA patterns and SEI architectural styles)?
- How do you apply architectural patterns when designing the architecture of a system? Is there a modeling language fit for this purpose? Can we identify common abstractions that can act as primitives for modeling architectural patterns?
- How are views related to architectural patterns? Can there be a one-to-one mapping? Or is it rather a N-to-M mapping?
- Which engineering techniques can be applied to find, select, and apply architectural patterns? How can a design decision based on patterns be found systematically?

The goal of the focus group was to examine how practitioners, researchers, developers, etc. make use of the rich but heterogeneous and diverse collection of patterns in the literature. The participants were asked to reflect and discuss the above issues from their own personal experience.

2 Group Discussion

We started off by discussing the two concepts of architectural patterns and architectural views – with the goal to find insights on their relationship. All 11 participants of the focus group stated that they use architectural patterns in their daily practices. The use of patterns varied from limited use of a few patterns to a systematic use of pattern languages. On the other hand, the concept of architectural views is not as much in mainstream use as architectural patterns are. Only 4 out of 11 participants actually use architectural views in their daily practice and those concern mostly coarse-grained views. Views are widely considered by practitioners an abstract concept, valuable for classification purposes but not really useful in industrial practice.

Next, we discussed the experiences of the participants in the use of architectural patterns. It turned out that the practices varied widely:

- Some participants model architectural patterns explicitly in the architectural design.
- Some participants merely apply them in the system without explicitly modeling them.
- Some participants lie in the middle of the two aforementioned categories: they apply the patterns without modeling them, but they do represent them textually or informally, in order to communicate with the stakeholders.

In general the participants agreed on the danger to use patterns blindly, without aiming to address stakeholders concerns or architectural drivers. Patterns should be the *means* to an end, that is, to satisfy concrete requirements, and not a self-fulfilling prophecy. Therefore application of patterns is all requirements-driven and architects which are constrained by such requirements cannot choose patterns ad-hoc, or have patterns imposed on them. In other cases, some architects do not choose patterns at all, but the patterns just evolve from the architecture design iterations. The participants agreed that, independently on how patterns are applied in a system architecture, they are seen as an effective vehicle to transfer the architectural vision to the stakeholders.

The relationship between architectural patterns and views lead to a more controversial discussion. We based the discussion on the set of views from the paper *Architectural patterns revisited - a pattern language*¹. The majority of the participants did not use views in practice, but they did have some ideas about what stakeholder concerns or architectural drivers should be addressed by architectural views. The participants proposed different names for some of the views and the patterns and they generally agreed that such a view-based categorization of patterns is useful for presenting the patterns and relating them with each other. However, even though this categorization serves the purpose of portraying the pattern language in paper, the architects usually have their own sequencing of applying the patterns in practice. This sequencing could depend on the following dimensions: the specific application domain, the architect or architecture team, the architecting process used, the organization, and perhaps the development project itself. Therefore

¹P. Avgeriou and U. Zdun. Architectural patterns revisited - a pattern language. In Proceedings of 10th European Conference on Pattern Languages of Programs (EuroPlop 2005), Irsee, Germany, July 2005.

if we examine case studies of how patterns are applied, with respect to each one of these dimensions or a combination of them, we can come up with different categorizations of the patterns. Perhaps the most useful dimension is the application domain, because it could lead to pattern sequences reusable across the domain.

For the second half of the focus group we decided to discuss cases of software architectures that the participants had been involved in, with respect to which patterns they used and how they categorized them. The participants formed two groups to make the discussions more manageable. The result of the brainstorm that followed confirmed the thesis that was stated before: there can be several alternative categorizations of the patterns according to how exactly architects use them in a project. An example of such a categorization scheme, which also depicts the sequencing of actions performed by an architect is the following:

1. Fundamental Decomposition
 - (a) Design of fundamental elements/decomposition (vertical/horizontal): Layers, Components, Subsystems
 - (b) Refinement of the decomposition, things that describe entities and components: In-direction Layer, Database Access Layer, Repository
2. Communication and Interaction
 - (a) Communication Design: Container, Broker, RPC, Message Queues, Publish/Subscribe
 - (b) Invocation Design: Explicit Invocation, Implicit Invocation, P2P
 - (c) Interaction Design: Blackboard, Pipes & Filters, MVC
3. Satisfaction of other non-functional requirements such as performance, scalability, reliability: Microkernel, Reflection, Plug-in, Interceptor.

3 Conclusion

The practice of software architecting is becoming mainstream, but the application of architectural patterns is severely limited due to the lack of an effective pattern classification. The focus group concentrated on this topic and identified the current state of practice. There is much controversy on the pattern categorization since there is no single correct solution, but several categorizations for architectural patterns are possible according to the dimensions we look upon them. The view-based categorization seems to be useful for presenting the pattern language and understanding the patterns and their relationships, but not so much for applying the pattern in practice. We expect that in mature domains or in specific organizations, different categorizations will come up aligned with the pattern sequencing that is performed in practice. We will monitor such developments in the patterns community and possibly explore them in future workshops and focus groups.

4 Participants

We extend our thanks to the following experts, who participated in the focus group, sharing their valuable experiences and insights in this challenging and intriguing topic:

- Arno Haase, Arno Haase Consulting, Germany

- Keith Braithwaite, WDS Global, UK
- Thomas Mey, Münchener Rückversicherungs-Gesellschaft AG, Germany
- Lubor Sesera, SOFTEC, Slovakia
- Wolfgang Herzner, ARC Seibersdorf research GmbH, Austria
- Frank Buschmann, Siemens AG, Germany
- Jorge Luis Ortega-Arjona, Facultad de Ciencias, UNAM, Mexico
- Markus Voelter, Independent Consultant, Germany
- Klaus Marquardt, Dräger, Germany
- Andy Longshaw, Blue Skyline Ltd., UK
- Amir Raveh, Motorola Inc., Israel