Abstract. The aim of the fourth workshop on Object-Oriented Architectural Evolution was to discuss in more detail a number of important issues raised during the previous workshop: the relationship between domain analysis and software architecture, the importance of architectural views and layering techniques, and the applicability of existing object-oriented principles and evolution techniques. This paper summarises the results of the debates held about these issues, reports on convergences of view taken place during the workshop, and suggests some research topics that are worthwhile to pursue in the future.
2 Workshop Preparation

2.1 Q & A-Style

The nature of the workshop was intended to be incremental, building further on the results of the last three years. To ensure an active collaboration between the participants, the call for participation was built up in a Q & A-style. After a briefing of the results of the previous workshop [3], a number of tentative open questions was suggested to which participants needed to provide an answer before the workshop.

We are convinced that this way of soliciting submissions greatly contributed to the success of the workshop. The process of asking questions is a well-known hermeneutic cognitive process in philosophy. It made it much easier to compare points of divergence between participants’ opinions, and enabled us to detect points of agreement on particular topics. Therefore, we believe that a Q & A-style would be an interesting alternative for other workshops as well.

2.2 Categories of Questions

Since the previous workshop [3] emphasised the need for domain analysis, as well as the importance of architectural views in combination with a layering mechanism, we decided to raise specific questions with the aim to explore these issues in more detail. The questions were subdivided into 5 categories:

1. Domain analysis
   a) What is the precise relationship between domain modelling and architectural design/modelling?
   b) How can domain analysis be used to derive a better (i.e. less change-sensitive) software architecture?
   c) Can we predict certain types of architectural evolution based on a given domain analysis? Which ones? How?

2. The use of multiple architectural views
   a) Should there be a predefined set of architectural views, or do the kinds of views depend on the problem domain?
   b) Is there a relationship between the different architectural views? Should we allow for explicit constraints between the views? How? Why (not)?
   c) Is there a correspondence between the architectural views and the architectural styles that can be used in those views?

3. Layered approach
   a) How should the different architectural layers be related? Should we put explicit constraints between them? How?
   b) Should there be a limited set of layers depending on the architectural view taken, or can there be an unlimited number of layers?
   c) How can layering ease the transition from a software architecture to the (object-oriented) software implementation?
   d) (How) can architectural styles other than a layered one be used to (i) facilitate evolution; (ii) ease the transition to the software implementation?