The building block method. Component-based architectural design for large software-intensive product families
Müller, J.K.

Citation for published version (APA):

General rights
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), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
The Building Block Method

Component-Based Architectural Design for Large Software-Intensive Product Families

Jürgen K. Müller
1. The system-theoretic view of software engineering is more powerful than the development-oriented view often used in practical SW engineering (chapter 2).

2. Object design, aspect design and concurrency design are three essential dimensions in the design of embedded systems (chapters 3, 4, 5, 6).

3. Qualities and aspects are both orthogonal multi-view descriptions of a technical system. Qualities are for describing a system from an external perspective orthogonal to its (functional) features while aspects are for describing the functionality of a system orthogonal to its constituting components (chapter 5).

4. The promise of reusing software components does not come true without an overall aspect design. Aspects are the backbone of a component-based software architecture (chapter 5).

5. Component frameworks, platforms and product family architectures are foremost the result of consolidation and not of innovation (chapters 7, 8, 11).

6. Scaling up a solution is much more difficult than scaling down a solution. Nevertheless, in both cases a check for the relevance of the solution to the problem has to be performed.

7. Due to a lack of understanding of both their customers and developments in essential technologies, companies are often unable to predict even the near-term progression in product features.

8. Creative methods do not exist. Good methods are consolidations of good as well as bad experiences.

9. Organisations without a technical hierarchy produce democratic system designs.

10. Discussions are an important cultural phenomenon in German and in Dutch organisations. However, their function is quite different. In Dutch culture they serve to achieve harmony within a group, while in German culture their function is to remove uncertainty.

11. The application of divide and conquer leads to divide and lose.