



Software Architecture for Product Families: Principles and Practice

By Mehdi Jazayeri, A. C. M. Ran, Frank Van Der Linden, Alexander Ran

Download now

Read Online ➔

Software Architecture for Product Families: Principles and Practice By Mehdi Jazayeri, A. C. M. Ran, Frank Van Der Linden, Alexander Ran

Software development organizations are now discovering the efficiencies that can be achieved by architecting entire software product families together. In *Software Architecture for Product Families*, experts from one of the world's most advanced software domain engineering projects share in-depth insights about the techniques that work -- and those that don't. The book offers a solutions-oriented, case-study approach covering the entire development lifecycle, based on advanced work done by three of Europe's leading technology companies and their academic partners. Discover the challenges that drive companies to consider architecting product families, and the new problems they encounter in doing so. Master concepts and terms that can be used to describe the architecture of a product family; then learn how to assess that architecture, and transform it into working applications. The authors also present chapter-length, real-world case studies of domain engineering projects at Nokia, Philips, and ABB.

↓ [Download Software Architecture for Product Families: Princi ...pdf](#)

📄 [Read Online Software Architecture for Product Families: Prin ...pdf](#)

Software Architecture for Product Families: Principles and Practice

By Mehdi Jazayeri, A. C. M. Ran, Frank Van Der Linden, Alexander Ran

Software Architecture for Product Families: Principles and Practice By Mehdi Jazayeri, A. C. M. Ran, Frank Van Der Linden, Alexander Ran

Software development organizations are now discovering the efficiencies that can be achieved by architecting entire software product families together. In *Software Architecture for Product Families*, experts from one of the world's most advanced software domain engineering projects share in-depth insights about the techniques that work -- and those that don't. The book offers a solutions-oriented, case-study approach covering the entire development lifecycle, based on advanced work done by three of Europe's leading technology companies and their academic partners. Discover the challenges that drive companies to consider architecting product families, and the new problems they encounter in doing so. Master concepts and terms that can be used to describe the architecture of a product family; then learn how to assess that architecture, and transform it into working applications. The authors also present chapter-length, real-world case studies of domain engineering projects at Nokia, Philips, and ABB.

Software Architecture for Product Families: Principles and Practice By Mehdi Jazayeri, A. C. M. Ran, Frank Van Der Linden, Alexander Ran **Bibliography**

- Sales Rank: #5259968 in Books
- Published on: 2000-01-15
- Original language: English
- Number of items: 1
- Dimensions: 9.50" h x 7.75" w x .75" l,
- Binding: Hardcover
- 288 pages

 [Download Software Architecture for Product Families: Princi ...pdf](#)

 [Read Online Software Architecture for Product Families: Prin ...pdf](#)

Editorial Review

From the Inside Flap

This book presents a view of software architecture for product families. It is addressed to researchers, engineers, and managers involved in the development of software-intensive products. The book is based primarily on the experience gained in the project ARES (Architectural Reasoning for Embedded Systems), carried out by six partners, funded by the European Commission. The project was concerned with the application of software architecture research results to practical problems faced by the industrial partners.

The three industrial partners shared similar problems in coping with the management of software product evolution in the presence of increasing feature requirements and decreasing time-to-market requirements. Software architecture, particularly applied to product families, appeared to offer some solutions, but there were many approaches and methodologies from which to choose. The goal of the project was to evaluate promising software architecture research results by applying them to some typical software evolution problems in industrial settings. One of the mottos of the project was "industry as laboratory."

The project was carried out by three industrial and three university partners:

Nokia Research Center, Philips Corporate Research, Asea Brown Boveri (ABB) Corporate Research, Technical University of Vienna, Imperial College of Science and Technology (London), and Polytechnic University of Madrid. During the project, we produced many results that had impact on the industrial partners (in terms of practice) and the university partners (in terms of curriculum development), as well as a large number of publications and presentations in scientific conferences and journals. Rather than collecting the many results already presented in journals

and conferences reporting on progress in the ARES project, we decided to integrate the results gained in the project and present a comprehensive view of the approaches we followed, of what worked and what didn't.

The purpose of this book is to bring some of the salient results together to make the big picture accessible to a wider audience. In short, the book attempts to give the reader the perspective gained by the project partners after three years of experience in applying research results to practical problems of software architecture for software-intensive systems. All chapters have been written specifically for this book.

The Introduction introduces the background of the project and the case study-based approach that we chose for running the project. It explains how the project came about and how we decided on the particular foci for the project.

Chapter 1 presents a conceptual framework for software architecture. One of the problems the field of software architecture faces is the lack of a widely accepted terminology or even a set of concepts. In this chapter we present the major concepts and define supporting terminology that allows us to discuss the problems and solutions in a uniform way. Chapters 2 through 4 are the "technology" chapters, each dealing with one major aspect of software architecture for product families: architecture description, architecture assessment, and architecture recovery.

Then follow three "experience" chapters. Chapter 5 presents the experience of Philips in applying software architecture in consumer electronics. Chapter 6 presents several experiments carried out at Nokia to deal with the documentation and assessment of software architecture. Chapter 7 describes ABB's experience in developing a software architecture for a family of train control systems, starting with three existing systems. The experience chapters describe the case studies in detail. Each chapter describes the state of the art at the start

of the project in the particular area, the problems we observed and tried to address in that area, and some of the ARES results.

Software architecture, product lines, and product families are all active areas of research and the subject of intense interest in industry. There are already several good books available that present different views of software architecture both for practitioners and researchers. Some present specific methodologies, some present research approaches, some present practical experiences. Our book may best be characterized as describing the application of research results to practice. Its aim is to show the reader how current research results may be applied and how to overcome the inevitable shortcomings of research results in practice.

For example, Chapter 2 introduces the concepts and approaches in software architecture description and concentrates mostly on the language Darwin, which was invented at Imperial College, one of the academic partners in the project. Chapter 5 describes how Darwin was applied at Philips and why the engineers at Philips had to modify Darwin to use it in practice. Indeed, they ended up developing their own tool, Koala, borrowing many of the concepts of Darwin but not the language itself. Chapter 7 describes what happened when ABB tried to use Darwin in architecture recovery. It worked for some things but not for others. ABB also had to do its own customization. We expect these kinds of results to be of interest to people concerned with software architecture both in academia and in industry.

Chapter 8 concludes the book by providing a high-level summary of the results and an outlook for future developments in the area of software architecture. It also discusses the critical issue of technology transfer in such a rapidly evolving field.

This is not the definitive work on software architecture families. No such

book exists and will not exist for several years to come. But we need different views of the problem until eventually there will be enough understanding to have a textbook. This book is a modest step in that direction.

How to Read This Book

The project ARES started in December 1995 and was completed in February 1999.

During this time, many people from all the partners worked on the project--some for a limited period of time and others during the whole period. The focus of the project was to produce results that could be exploited immediately by the partners.

For example, we at the Technical University of Vienna started a software architecture course relying heavily on the case studies from the ARES project. At Philips, engineers developed new tools for documenting software architectures and automatically generating code for module interconnection. At Nokia, engineers developed Web-based tools for organizing the documentation of complex systems. At ABB, the results of some of the ARES work drove the design of a new family architecture for train control systems. At Imperial College, the feedback from engineers in the three companies led to improvements in the Darwin tool set. At the Polytechnic University of Madrid, the researchers developed techniques for architecture assessment and response-time analysis that were used by the different industrial partners.

Although such results of the ARES project were useful and beneficial to the partners, the goal of the project was much larger. As a research project, we wanted to communicate our results to the larger research and industrial community.

This we did through many publications in conferences and journals. Toward the end of the project we asked ourselves if there was anything else left to say that our publications did not say. The answer was that indeed much was left unsaid. The publications for the most part presented technological solutions to particular problems encountered at the time of publication. What was missing

was any kind of integration--a larger conclusion, a bigger picture based on more than three years of experience. We decided that a book with the focus on the larger picture was the answer.

Rather than concentrating on point solutions to specific technical problems, the book would give a perspective of a three-year experience. Each chapter tries to look back three years and move forward to share a three-year perspective with the reader. This was the best way we could imagine to communicate what we learned in the project. This kind of presentation in each chapter gives the reader the opportunity to see the whole picture: our assumptions, our starting point, and where we ended up. As a reader, you can judge how our experiences may or may not apply in your context.

The project investigated different aspects of software architecture. In Chapter 1, "ARES Conceptual Framework for Software Architecture," Alexander Ran tries to abstract and summarize from these different aspects the project's collective understanding of an architecture-oriented approach to software engineering.

In some ways, this chapter is the main result of the ARES project to be communicated to the community. Experienced software engineers and architects may read this chapter immediately and identify with many points it makes. Less experienced engineers probably will find this chapter unconvincing or hard to read initially.

They will be better off reading the chapter quickly at the beginning and then returning to read it more carefully after they have read the rest of the book.

The rest of the book contains three "technology" chapters and three "experience" chapters. The technology chapters (2 through 4) abstract the technological aspects of our experience in a particular area. For example, Chapter 4 ("Software Architecture Recovery") tries to organize the processes that we used in applying architecture recovery to the ABB case study. It is not a research presentation as much as an attempt to document the approaches that worked in the case study. But rather

than presenting the details of the case study, it presents methodological approaches that may be useful in other case studies of interest to the reader. Thus, in reading one of the experience chapters (5 through 7), you can form your own conclusions about the experiment and the experience. A technology chapter presents our conclusions about the applicability of technology to real-world situations. The technology and experience chapters may be read in any order if the reader prefers to jump to a favorite subject.

0201699672P04062001

From the Back Cover

--Paul Clements, Software Engineering Institute

"The authors present a very knowledgeable presentation about software development and software architecture. There are good techniques here that would interest other software architects and designers." -- Henry McNair, Hewlett-Packard

The fast pace of modern business is creating new challenges for software engineers. These professionals are faced with the daunting assignment of delivering more robust software for less money--and in less time. In addition, many must now learn to efficiently develop software for a suite, or family, of products. Research has shown that a product-line approach to creating software can be achieved, but where does a software practitioner turn for information and guidance on such a major, potentially difficult transition?

Using an example-based approach, *Software Architecture for Product Families* provides the software engineering community with a valuable resource for devising and, more importantly, implementing a successful product-family architecture. The concepts and technologies set forth in this book are based on the authors' cutting-edge experiences with the ARES (Architectural Reasoning for Embedded Systems) Project, a major European research initiative that proved that software reuse can be attained more easily through a product-family approach.

This book simplifies the intimidating process of learning how to design software for product-line engineering by focusing on three key aspects of architectural practice: architecture description, assessment, and recovery. Practitioners will learn how to pinpoint the source of their problems, focus on the big picture, and see what kinds of problems can be addressed with the product-family approach. With this book as a guide, software developers will be able to implement a sound product-family architecture that will result in better quality, reduced costs, and decreased time to market. 0201699672B04062001

About the Author

Mehdi Jazayeri is a professor of computer science and heads the Distributed Systems Group at the Technical University of Vienna. He has extensive experience in industry, and has held both technical and management positions at Hewlett-Packard Laboratories, Synapse Computer Corporation, and TRW Vidar. He has co-authored textbooks on software engineering, programming languages, and software processes. Alexander

Ran is a Principal Scientist of Software Architecture at Nokia Research Center, where he leads research projects, provides consulting, and conducts architectural assessments of products for Nokia Business Units. Alexander is a contributor to the first and second PLoP books. He served as program co-chair of the First Working IFIP Conference on Software Architecture and the First Software Product Lines Conference. He was an invited speaker at the International Conference on Software Engineering (ICSE) and a keynote speaker on product families at the Symposium on Software Reusability. Frank van der Linden has been a researcher with Philips Research Laboratories for 15 years. Frank has broad experience in the component architecture of diverse electronic consumer and professional products. He recently moved to Philips Medical Systems, and is the editor of the proceedings for the special workshop on Software Architecture for Product Families being offered at ICSE. 0201699672AB04062001

Users Review

From reader reviews:

Brenda Lee:

The book Software Architecture for Product Families: Principles and Practice can give more knowledge and information about everything you want. Why must we leave the great thing like a book Software Architecture for Product Families: Principles and Practice? A few of you have a different opinion about publication. But one aim in which book can give many data for us. It is absolutely suitable. Right now, try to closer together with your book. Knowledge or data that you take for that, you could give for each other; you may share all of these. Book Software Architecture for Product Families: Principles and Practice has simple shape but you know: it has great and big function for you. You can search the enormous world by start and read a publication. So it is very wonderful.

Sonia Cramer:

Does one one of the book lovers? If yes, do you ever feeling doubt when you find yourself in the book store? Try to pick one book that you just dont know the inside because don't determine book by its cover may doesn't work is difficult job because you are scared that the inside maybe not as fantastic as in the outside search likes. Maybe you answer might be Software Architecture for Product Families: Principles and Practice why because the fantastic cover that make you consider about the content will not disappoint you. The inside or content is usually fantastic as the outside or cover. Your reading 6th sense will directly make suggestions to pick up this book.

Eric Sanders:

This Software Architecture for Product Families: Principles and Practice is great guide for you because the content that is full of information for you who all always deal with world and possess to make decision every minute. This book reveal it data accurately using great plan word or we can point out no rambling sentences in it. So if you are read that hurriedly you can have whole facts in it. Doesn't mean it only will give you straight forward sentences but hard core information with lovely delivering sentences. Having Software Architecture for Product Families: Principles and Practice in your hand like obtaining the world in your arm, info in it is not ridiculous just one. We can say that no publication that offer you world inside ten or fifteen moment right but this e-book already do that. So , it is good reading book. Hey Mr. and Mrs. busy do you still doubt in which?

Kenneth Armstrong:

In this era which is the greater man or who has ability in doing something more are more special than other. Do you want to become considered one of it? It is just simple way to have that. What you need to do is just spending your time little but quite enough to experience a look at some books. One of several books in the top collection in your reading list will be Software Architecture for Product Families: Principles and Practice. This book which is qualified as The Hungry Inclines can get you closer in turning into precious person. By looking way up and review this book you can get many advantages.

Download and Read Online Software Architecture for Product Families: Principles and Practice By Mehdi Jazayeri, A. C. M. Ran, Frank Van Der Linden, Alexander Ran #VB8RDO0KM9J

Read Software Architecture for Product Families: Principles and Practice By Mehdi Jazayeri, A. C. M. Ran, Frank Van Der Linden, Alexander Ran for online ebook

Software Architecture for Product Families: Principles and Practice By Mehdi Jazayeri, A. C. M. Ran, Frank Van Der Linden, Alexander Ran Free PDF d0wnl0ad, audio books, books to read, good books to read, cheap books, good books, online books, books online, book reviews epub, read books online, books to read online, online library, greatbooks to read, PDF best books to read, top books to read Software Architecture for Product Families: Principles and Practice By Mehdi Jazayeri, A. C. M. Ran, Frank Van Der Linden, Alexander Ran books to read online.

Online Software Architecture for Product Families: Principles and Practice By Mehdi Jazayeri, A. C. M. Ran, Frank Van Der Linden, Alexander Ran ebook PDF download

Software Architecture for Product Families: Principles and Practice By Mehdi Jazayeri, A. C. M. Ran, Frank Van Der Linden, Alexander Ran Doc

Software Architecture for Product Families: Principles and Practice By Mehdi Jazayeri, A. C. M. Ran, Frank Van Der Linden, Alexander Ran Mobipocket

Software Architecture for Product Families: Principles and Practice By Mehdi Jazayeri, A. C. M. Ran, Frank Van Der Linden, Alexander Ran EPub