Introduction – A Guided Tour of EuroPLoP 2007

Welcome. These are the proceedings of EuroPLoP 2007, the 12th European conference on Pattern Languages of Programs, which took place between the 4th and 8th July 2007 at Kloster Irsee in the heart of Bavaria.

What is presented here is not a record of the papers submitted prior to the conference, or even updated forms delivered at the conference itself. Rather, the papers here were part of an ongoing process that started in January 2007 (or before) and continued until January 2008. Many of the papers are part of an even longer process, as patterns mature over years and pattern languages grow out of a few core patterns.

The proceedings represent the results of a lot of work on the part of the authors of the papers who mine and document the patterns, on the part of the shepherds who help to refine the patterns for submission, and on the part of the workshop participants who help to temper the patterns in the metaphorical forge of the workshop. The papers are work-shopped in traditional PLoP style with all workshop participants giving feedback on each paper, and the authors sitting apart and absorbing the comments, criticism and praise. This very intense form of feedback is possible because of the atmosphere of mutual trust built during each conference, backed up as it is by friendship and respect gained by the EuroPLoP community over 12 years of conferences. The tranquillity of the location is a counterpoint to both the intensity of the workshops and the energy of extra-curricula activities.

Since the common interest of EuroPLoP participants is to exchange best practices, it covers a wide range of topics from best practices for social interaction through best practices for software architectures to patterns for embedded systems. This spectrum from the "soft" social sciences towards "hard" engineering domains contributes to the special character of EuroPLoP. Although different in their scientific background, the participants reach a level of exchange that stimulates creativity and bridges various fields. Managers interact with hardware designers, pedagogues question the clarity of software designs, and programming language experts challenge practices of HCI designers. Together with an intensive exchange between experts in the same domain, the inter-disciplinary dialogues raise questions that lead to the innovative results collected and conserved in this volume.

These proceedings contain 40 pattern papers out of the 51 pattern papers that were submitted to the conference. This high acceptance rate distinguishes EuroPLoP from other conferences in the scientific field. At a first glance, readers may fear that such a high acceptance rate is an indicator for low quality content. But the opposite is the case: instead of simply rejecting the authors' works, EuroPLoP has managed to develop a unique culture of supporting researchers and practitioners who collaborate to improve quality instead of competing to stay unique. Papers
that enter the EuroPLoP review process with average quality are taken up by pattern and domain experts who challenge the author in order to find the golden nuggets in the author's work. This process, called shepherding, lasts for several months and culminates in on-site discussions about the work in so-called writers' workshops.

There were six writers’ workshops at the conference. In addition to the papers in the writers’ workshops, there were four papers submitted to the writing group. These were considered to have potential that, with one-on-one, face-to-face shepherding at the conference they could progress sufficiently to enter a writers’ workshop by the end of the conference. After the writers' workshops, another round of improvement began that resulted in the 40 papers contained in this volume, all of them with a significantly improved quality compared to the time of initial submission and some really reached superior quality.

The proceedings of EuroPLoP 2007 are organized in a way that reflects this dialogue between social and technical aspects. All papers create an interesting sequence of pattern clusters that illustrates the multiple subjects for which patterns have been developed so far.

The first seven papers are patterns for various social interaction scenarios. In Fießer's paper, the processes of film making are discussed, while Kauhanen et al. present patterns that support the process of story authoring. Schmolitzky brings in his expertise from teacher–student interaction with a special focus on combining traditional lectures with hands-on experiences. Haase and Miedl as well as Schümmer and Tandler both look at patterns for meeting support. The collection of Haase and Miedl extends their meeting pattern language presented at EuroPLoP 2005. Schümmer and Tandler take a different approach by focusing on how computer technology can support meeting facilitation. Finally, the pattern language by Raveh and Homsky investigates the social glue that is required to build online communities, and the writing group paper of Chaturvedi reports on a strategy for team development.

The cluster of business, processes, and services discusses questions of business strategies and services as well as questions that are specific to software development projects. Weiss takes a look at competition and alliances between business partners in his paper with the provocative name "In Bed with the Enemy". Kelly also looks at business strategies, for meeting the customers' needs and identifying the services requested by the customer. Holtel addresses the same question from a software developer's perspective and provides hints on how to focus the development process of the desired services. Henrich's and Zdun's paper answers the question about how such services can be invoked from business processes. Clearly, the processes also need to be well structured, which is the focus of Laue and Gruhn's paper that gives "good and bad excuses for unstructured business process models".
Other patterns in this section are focusing on specific aspects in collaborative software development. Salecker describes best practices for using configuration management approaches. One recommendation given by Salecker is to validate all code at check-in time in order to raise the quality of the product. Abath Neto and colleagues were also concerned with product quality while writing their pattern language on scripting acceptance tests.

Media and interaction designers met with software developers when it came to focus on patterns addressing **Human-Computer-Interaction** aspects. Three papers investigated aspects of media presentation: Kohls and Windbrake collected practices for shaping interactive graphics that are used in educational software to communicate dynamic content. The writing group paper of Zimmermann and colleagues addressed the important issue of accessibility, e.g., the question of how video material can be more accessible to deaf people. Rüping continued his work from previous years on web content management. His contribution situated the patterns of three previous pattern languages and combined them in a big picture, a trend that can be found in many application areas of patterns when the pattern language slowly matures. Lukosch and Schümmer extended their pattern language on Computer-Mediated Interaction with some patterns for designing the login process in collaborative applications.

Schadewitz and Jachna moved the participants' attention to the cross-cultural issues in design activities. They observed the interaction between students in Hong Kong and students in Austria as well as students in Hong Kong and students in Korea and found interesting issues that should be considered when designing collaboration infrastructures for cross-cultural interaction. This paper actually bridges the gap to the contributions collected in the social interaction cluster. Not only the paper but also the way how it was discussed at EuroPLoP was a novelty for the conference. Since Nicole Shadewitz was unable to attend the conference in person, she participated in the writer's workshop as a virtual participant. Using a live video link, she was virtually "placed" on a chair in the circle and was successfully integrated into the discussion process of all papers of the workshop.

One of the core topics of EuroPLoP has always been **software architectures**. The paper by Siddle takes a retrospective look at existing patterns and shows how the combination of patterns in pattern sequences can unleash the full power of the pattern approach. Marquardt discusses how boundaries between components interrelate with organizational structures and provides guidelines for overcoming artificial boundaries in the design. The paper is a good example how organizational questions again interrelate with engineering practice and shows that software construction is only partially a technical task. Schütz addresses the problem of variability in product line engineering, again a problem space that has implications on the software development process as well as on the system architecture. One concrete example of configurability can be found in the
writing group paper of Meffert and Phillipow who address the problem of configuring threaded applications.

Several papers focused on a specific aspect of most software architectures: **database access** and **object management**. Böhm and colleagues presented a paper that explains how data resources can be substituted or used alternatively at runtime of a system. Welicki provides three patterns for bridging the object-oriented paradigm with relational databases in the data access layer of enterprise applications. In the same domain, Wellhausen collected best practice for improving the performance of this bridge by allowing developers to explicitly control the prefetch behavior of individual database queries. De Rore and Snoeck identified best practices for reconciliation, a process implemented to detect and report the inconsistencies that can appear between two datasets that represent the same information, in particular between a master and a slave application. This is especially important to detect and avoid inconsistencies between replicas of the same domain data stored in different (legacy) databases. Another source for inconsistencies is given by a changing context. The modeled data representation becomes inconsistent with the real concepts represented by the model. Wegner and Marti collected patterns supporting the restructuring of attributes serving as classification categories, so-called dimensions. Bienhaus was also interested in mapping reality to a data model. His pattern language for unique product identification takes up a problem that is common to manufacturing, namely that real parts in the construction line have to be identified and mapped to their digital counterpart.

This pattern language already opened up the space for hardware-oriented patterns. For several years, EuroPLoP has been a forum for discussing patterns in the field of **embedded systems** and **network technology**. Pont and colleagues have conducted research in the development of reliable embedded systems and collected a large set of patterns for such systems within the last decade. The time-triggered scheduler presented in this paper extends this collection and sheds light on the question when this scheduling strategy would be appropriate. Another extension to the pattern language is the paper by Wang and colleagues that takes a closer look at shared resources in time-triggered scheduling. Bammi addresses a comparable context with his pattern collection for data acquisition in embedded systems that can, e.g., be used when designing analog-digital sampling components. Ortega Arjona took a look at problems that frequently arise when developing parallel programs. His patterns provide an overview on best practices for structuring the communication between the parallel processing components, be it parallel processes, different hardware units or networked computers in a distributed system. In the context of networked applications, two papers were discussed at EuroPLoP: Fernandez and colleagues extended their work on security patterns by investigating mechanisms for secure wireless communication. From
the same group, Pelaez and colleagues contributed a pattern language that describes the structure and application of Voice over IP signaling techniques.

Last but not least, EuroPLoP has been a forum for discussing fundamentals of **Programming Languages and Aspects**. Haase introduces the reader to the craft of language design, e.g., for contexts where a domain-specific language is required. Côté and colleagues investigated the role of problem frames initially proposed by Jackson and considered them as patterns that classify software development problems and can be referred to as "problem patterns". Three papers addressed issues in the area of aspect-oriented programming. Noble and colleagues collected patterns that can serve as the beginnings of a taxonomy of aspect-oriented design. They present problems that are suitable for an aspect-oriented solution and provide guidance on designing an appropriate aspect. Bartsch and Harrison compared object-oriented implementations of design patterns with aspect-oriented implementations to identify differences in coupling. Schmidmeier showed how an aspect-oriented paradigm is reflected in the software development process and thereby closes the loop to the beginning of the EuroPLoP proceedings where social aspects of software development were considered.

The final part of the serious business at EuroPLoP consists of focus groups. Focus groups are interactive workshops that explore a particular theme. The proposals are submitted and reviewed at the same time as the pattern papers. Six Focus Groups were accepted and took place during the conference, of which two have submitted a report for these proceedings. The Focus Group **AJAX The big new thing or the big new bubble** set out to explore the substance of this new technology. The groups identified benefits and problems with the technology, with the group’s confidence level for each. They concluded with real benefits especially for users but also problems for the developers while this technology is maturing. The Focus Group **That works for me!** discussed personal experience with various software development practices, and the context in which they worked or did not work. The group looked for patterns of contexts where practices or methodologies work. They found contexts where practices will not work, and that no context can guarantee the success of a project.

As a counterbalance to the intense work involved in the conference, EuroPLoP has many organized activities ranging from the fun and challenging games led by the official conference game master: Querdenker George Platts. George also runs an evening art studio with active participation from conference attendees and music/video productions not least during our traditional banquet on Friday evening. Tours of the monastery and its associated brewery are other popular activities. The goings on at the conference are reported in the daily conference newspaper – Kloster Hearsay – and discussed in the bar or in the monastery garden.
As ever, EuroPLoP 2007 was an enjoyable experience. If you contributed to it in any way then we thank you. If you did not, then please come and explore the next EuroPLoP conference (http://www.europlop.org) and consider joining the community.

We hope that these proceedings are helpful not only as a record of the inspiring discussions at EuroPLoP but also as a source for improving your practice. And we hope that you will enjoy reading the following pages.

Lise Hvatum, Programme Chair
Till Schümmer, Conference Chair