Lecture Notes in Computer Science 7365

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Services Science

Subline of Lectures Notes in Computer Science

Maritta Heisel (Ed.)

Software Service and Application Engineering

Essays Dedicated to Bernd Krämer on the Occasion of His 65th Birthday

Volume Editor

Maritta Heisel
Universität Duisburg-Essen
Fakultät für Ing.-wissenschaften
Abteilung INKO
Fachgebiet Software Engineering
47057 Duisburg, Germany
E-mail: maritta.heisel@uni-duisburg-essen.de

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-30834-5 e-ISBN 978-3-642-30835-2
DOI 10.1007/978-3-642-30835-2
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2012938718

CR Subject Classification (1998): F.3, D.2, F.1, H.3, H.4, C.2

LNCS Sublibrary: SL 2 – Programming and Software Engineering

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Bernd J. Krämer
©Veit Mette, Bielefeld

Preface

This volume appears on the occasion of Prof. Dr. Bernd Krämer's 65th birthday. Close scientific companions, who have worked with Bernd for many years, contributed to this volume. The papers reflect the remarkable span of Bernd's own research. Bernd's career started with research in formal languages to enable parser generation for programming language definitions with van Wijngaarden grammars, which became popular with Algol 68. Working in the Institute for Software Technology at GMD (Gesellschaft für Mathematik und Datenverarbeitung mbH Bonn) in the direct neighborhood of the Institute for Information Systems Research led by Carl Adam Petri, he soon became interested in Petri nets as adequate means to model distributed software systems. Together with his colleague Heinz Schmidt he developed a specification and modeling language, SEGRAS, combining high-level Petri nets and algebraic specifications. These works led to a large ESPRIT project (1983-1988) aiming at the development of a software engineering environment for non-sequential software systems. With this project he widened his research activities to software engineering, in general, taking into account distributed and safety-critical systems, in particular. This was the context in which I met Bernd, who has been an important mentor for me since then. He not only served as a reviewer for my habilitation thesis, but was always there to support me in my academic career and encouraged me during the long road to becoming a professor for software engineering myself. His experience of teaching at a distance learning university inspired Bernd's later work on e-learning. Following the evolution of the field, he also turned his attention to such important and highly promising research fields as middleware, component-based software engineering, service-oriented computing, and, more recently, social computing.

The articles contained in this volume reflect most of Bernd's research themes. They cover Petri nets and theoretical computer science, software and service engineering, cloud computing, and e-learning.

Wolfgang Reisig discusses various extensions of Petri nets, concerning, for example, fairness, hot and cold transitions, and distributed runs. Such extensions aim at not only extending the expressive power of Petri nets, but at the same time retaining the well-established analysis techniques coming with Petri nets.

Karsten Gabriel and Hartmut Ehrig use algebraic high-level (AHL) nets and their processes to model communication platforms and evolutions of such platforms. Algebraic high-level nets combine Petri nets and algebraic specifications. The corresponding processes model concurrent firing behavior of AHL nets. Gabriel and Ehrig use the example of the Apache Wave communication platform – together with a possible evolution – to demonstrate the power of AHL nets and processes and to illustrate their formal properties.

Herbert Weber discusses the role of modeling in engineering. He proposes *profiling* as a methodology to describe computing artifacts and their application environment along different axes, namely, abstraction, granularity, and virtualization. The methodology consists of three stages, where the expressiveness of the notation is increased gradually.

Marco Konersmann and Michael Goedicke present a conceptual framework that serves to link software architectures and their implementation. They define an intermediate language to which different architectural description languages can be mapped. The resulting architecture descriptions are then embedded into the component model used to implement the architecture by means of code patterns. Since bi-directional mappings are defined between the intermediate language on the one hand and architectural description languages or component models on the other hand, changes in one description can be propagated into the other description. Thus, architectural descriptions can be kept consistent with their implementation.

Djamshid Tavangarian gives an overview of virtualization techniques and shows how virtualization can be used to optimize power consumption of computing centers and to enhance security in cloud computing.

Frank Leymann's contribution builds on the concepts of compute units and social compute units. Compute units specify the structure of an application as well as plans needed to manage it. Social compute units consist of a compute unit, augmented with people who have some relation to the software, for example, they created it. Leymann proposes to enhance (social) compute units with data sets serving as input or output of the compute unit, which results in *linked (social) compute units*. Such linked compute units support scientific work based on computer simulation or software-based data analysis.

Kristian Beckers, Stephan Faßbender, Maritta Heisel, and Rene Meis propose patterns for establishing the context of SOA (service-oriented architecture) systems. These patterns represent the different layers making up such an architecture as well as the relations between those layers. They also structure the environment of SOA systems and identify relevant classes of stakeholders that are described using specific templates. The authors also present a method to instantiate the patterns, which can be embedded, for example, in secure software development life-cycles.

Michael Papazoglou addresses the problem of cloud federations. He proposes the concept of cloud blueprinting to make cloud computing more flexible and give cloud developers more influence on cloud-based applications. Cloud blueprinting makes it possible to combine different cloud services offered by different providers. Cloud developers can orchestrate and configure their applications using a *blueprint framework* that offers a variety of languages to support the approach.

Ingolf Krüger, Barry Demchak, and Massimiliano Menarini present the architectural definition language OpenRichServices (ORS). ORS allows one to define services as functions. It distinguishes application and infrastructure concerns and

comes with a deployment sub-language. Moreover, it allows for dynamic changes of the structure and behavior of the system.

Kirill Mechitov and Gul Agha show how cyber-physical systems (i.e., systems comprising networked physical devices) can be engineered. They propose a service composition-based software architecture that supports a network-wide programming model, sharing and re-use of resources, and late binding of services and resources. The architecture consists of two layers. On the knowledge layer, meta-actors and meta-objects control the execution process. On the operational layer, dynamically composed and deployed services are executed.

Christoph Rensing, Stephan Tittel, and Ralf Steinmetz present novel approaches to support authors and learners in location-based learning scenarios. In the first scenario, authors can generate location-based content with the help of their smartphone. In the second scenario, learners are provided with a semantic media wiki, which allows them to enter their notes on the spot.

Thus, the interested reader can find in this volume a rich fund of novel approaches and interesting discussions of various important research topics that have been inspired by Bernd Krämer's diverse research activities and contributions.

April 2012 Maritta Heisel

SDPS Conference 2001 in Pasadena, California

First Dagstuhl Seminar on Service-Oriented Computing (SOC), May 2005

The organizers of the symposium, R. Valk, B.J. Krämer, and W. Reisig (from left), with T. Petri holding the IEEE Computer Pioneer Award Carl Adam Petri received in 2009

Introductory Address

I am deeply honored and pleased to be able to write a short foreword – or "Grußwort" in German – to this very special volume – or "Festschrift" – that constitutes contributions written as a token of appreciation of the scientific community to one of its prominent members, Professor Bernd Krämer from FernUniversität in Hagen in North Rhein-Westphalia, Germany. The idea of a Festschrift, though primarily being characteristic of the German academic system, has also become popular in many other countries as people realize that such a book can play an important role as a sign of a natural need to appreciate a long service, and as a sign of continuity of the school founded by the person to whom the volume in question has been devoted.

This Festschrift is very important to all of us in the academic community. First of all, it honors a prominent scholar and scientist, one of those who have recognized early that issues related to broadly perceived information technology, software engineering, etc. will become key fields due to theoretically interesting and challenging problems to pose and solve, and to their vital practical importance. Professor Krämer has been one of the most active members of the academic communities in these fields, both as a teacher, researcher and mentor to his numerous students, a founder of new research groups that have quickly become leaders in their respective fields, editor of many journals, member of international program committees and invited speaker at numerous highly respected conferences, member of advisory boards of various bodies, etc. Second, the collection of contributors to this Festschrift indicates that he has really founded and established over the years a school in the best scholarly tradition that goes back to ancient times. Third, on a personal basis, I am happy to write these few words to honor a great human being and friend with whom I not only share a very similar type of engineering education and academic career but also research interests and views.

To summarize, I wish to express my deep appreciation for all that Professor Krämer has done for his university and his group and – more generally – for science and the entire research community. However, since looking into the future is always more interesting and fruitful, I think that his future works will continue to shape research both at his university and all over the world. His scholarly and research stature, vision and passion will continue to inspire all of us.

April 2012 Janusz Kacprzyk

A Professional Passion for Software Engineering

It is indeed a great pleasure to write a foreword for this book dedicated to Bernd Krämers on the occasion of his 65th birthday. Bernd Krämer has been a full professor of software engineering for many years and has made significant contributions to service-oriented computing such as Web-based hypermedia systems, component-based software engineering, technology for e-learning, distributed systems management and engineering, safety-related software, and formal methods. His dedication to software engineering research and practice significantly impacted the fundamental underpinnings of the field.

The term "software engineering" (SE) was first coined at a NATO conference in 1968. It was gradually accepted as an academic discipline after the establishment of *IEEE Transactions on Software Engineering*, and the International Conference on Software Engineering (ICSE) in 1975. Much, of course, has happened during the past 37 years as the field exploded in all directions. As we enter the age of ubiquitous computing, computing systems networked together constitute a tremendous amplifier of the human mind with the amplification factor growing at an ever-increasing pace. Software will be a core part of this formidable power that is destined to alter our world in a most fundamental way.

What SE is up against is keenly in line with how the human race will be addressing our coming grand challenges including poverty, health and health management, security, energy, transportation, sustainability, etc. The systems needed to tackle these problems are unprecedented in scale, with billions of lines of code, and of a mind-boggling complexity as far as the amount of data, number of routines, number of connections and interdependencies, and emergent behaviors, etc., are concerned. While SE alone cannot solve these major issues, software will definitely play a key role in finding the solutions to these problems not only as key components of the overall solution, but also in the form of basic infrastructure which supports the collaboration of various communities of different disciplines working together.

Most of these mega-problems are inherently "wicked" as we are dealing with complex systems of systems which undergo continuous evolution while being used; systems built from heterogeneous elements that are constructed for different purposes at different times; decentralized systems with no clear ownership, etc. The wickedness of these problems includes, for example, "problem formulation and its solution cannot be separated," "symptoms and causes cannot be distinguished," and "there are usually no rules to determine when a solution is complete," etc. Given the scale and complexity of these potential systems, it is essential to understand what the key underpinnings of SE are, as many of the users/developers will not be software professionals.

Historically, SE began as a computational approach to problem solving. The thinking process behind such an approach is usually referred to as "computational thinking," or how people formulate problems and solutions so that the solutions can effectively be carried out by one or more computing devices.

But good software should also have logical beauty by reducing the detailed reasoning needed to a doable amount by utilizing principles such as "separation of concerns," etc. Over time, software professionals learned that they must develop systems which are not only logically elegant, functional and within budget, but also emotionally appealing in terms of user experience. As such, "design thinking," or how people utilize their intuitive ability to recognize patterns and to construct ideas that have emotional meaning as well as functionality, has been an integral part of the SE consciousness.

As SE is required to tackle complex problems which are systems of systems, the boundary between SE and that of system engineering begins to blur. SE must deal with not only systems of ultra-large scale, but also emergent systems, such as Wikipedia, which are developed by groups of people collaborating together, each contributing to different aspects of the system with an evolving infrastructure, over time. As a consequence of this system-oriented evolution, SE must include "system thinking," or how to negotiate complex interrelationships and processes of change, as part of its core make-ups as well.

If we accept that these three kinds of thinking, namely, computational, design, and system, form the core of SE's approach to problem solving, we must agree that SE is transdisciplinary in nature. It seems that the next evolution of SE could be the harmonious merging of these three kinds of mental disciplines into the SE consciousness in order to take up the many great challenges facing the SE community in the years to come. In this regard, Prof. Krämer has been a major contributor via his involvement in transdisciplinary research and education. He is the editor-in-chief of the *Journal of Integrated Design & Process Science*, and has been the longest serving President of the Society for Design and Process Science (SDPS). It is through his efforts that I see SE in a much larger context.

I would like to conclude with a personal note, as I have known Prof. Krämer for nearly two decades. He is one of the nicest gentlemen I have had the pleasure of meeting. He is always on the lookout to help a colleague, and he provides a calm presence, with a shy smile, even in the context of a heated debate. He has gained a great deal of respect not only for his intellectual prowess being a great teacher, but also for his leadership in service. I sincerely congratulate Prof. Krämer on the publication of this commemorative volume.

March 2012 Raymond T. Yeh

The University Teacher and Researcher

Computer science is a relatively young specialist discipline as many people who have witnessed the rapid growth of this discipline will remember. Nowadays it is an integral part of the curricula framework of many universities. It has produced results, without which our information technology-driven world would not work.

Prof. Dr.-Ing. Bernd Krämer is one of the pioneers of this discipline. He has shaped and enriched it by his research. In 1992 he became a full professor and the head of the Chair of Data Processing Technology of FernUniversität in Hagen. A media-based university like the FernUniversität, naturally profits to a high degree and valuable way from the experience of such a national and international well-known and respected expert.

In the development of data management solutions there are two inseparable questions: Will the system be developed in the right way? And: Will the right system be developed? The first question is related to the technical functionality. The second question is focused toward usability and user requirements. Otherwise the program will not be used and is therefore without any value.

Bernd Krämer always has both questions on his mind. This can be shown clearly by "edu-sharing" as one of his key projects.

In 2002, the Deutsche Forschungsgemeinschaft (German Research Council) announced a nationwide call for the development of centers of excellence in two fields. In each field, only four projects from universities were funded. Edu-sharing was one of the winners in the field "Digital Text and Data Centre for Collecting, Securing and Providing of Digital Sources in Research and Education." In March 2012, the four project teams presented their results in a final workshop, which was organized by Prof. Krämer at the FernUniversität. It was a great success.

Edu-sharing is an open educational resources (OER) program for a networked administration of e-learning and e-knowledge content. Via open archives initiative interfaces, different knowledge archives can communicate with each other and exchange metadata. "Find what will help you," with these words Bernd Krämer himself once described the benefit of edu-sharing for an effective learning and working process. The benefit for our university is obvious. It is a major step forward that the large amount of e-learning and e-knowledge content can now be administrated in a systematic way and can be made re-usable for scientists and students.

As an open educational resource, edu-sharing can be used by all schools and universities interested as well as by enterprises. Via this system, for example, teachers can exchange, add on, improve, adapt and evaluate content. Or to bring it to a point: to learn from and with each other. The results can be made directly available to teachers and students. In North Rhine-Westphalia (NRW) this system is run on some of the public computer centers, which support all

schools in the country with IT-services and learning content. In the long term, 6,800 schools in NRW will be connected to the edu-sharing platform.

This is only one of many shining examples of the innovative power of Bernd Krämer.

Furthermore, a special commitment was made by him to young scientists. As one activity among many others, he is busy in the "International Cooperation in Ambient Computing Education (ICACE)," a program for a transatlantic exchange of students.

He is editor-in-chief and member of several editorial boards of high ranked international scientific journals. At FernUniversität, he is the editor of our ejournal *eleed*. Prof. Krämer chaired several international congresses and program committees and has often been invited as keynote speaker.

Besides this, his engagement for our university goes far beyond the scientific part. He has supported and is supporting the development of FernUniversität as a member on several boards, in the Senate, and presently on the Supervisory Board of our university appointed by the minister of science.

I appreciate Bernd Krämer as a scientist, as a colleague for many years and especially as a person. It is a great honor and pleasure for me to make a contribution to this commemorative publication for his 65th birthday.

April 2012 Helmut Hoyer

Table of Contents