Int J Software Informatics, Volume 5, Issue 1-2 (2011), Part I, pp. 1–6
International Journal of Software and Informatics, ISSN 1673-7288

E-mail: ijsi@iscas.ac.cn
http://www.ijsi.org
Tel: +86-10-62661040

Preface

Foundations and Practice of Systems and Software Engineering

— Towards Scientific Contribution of Prof. Manfred Broy

Martin Wirsing[1] and Ruqian Lu[2]

[1] (Ludwig-Maximilians-Universität München, Germany)

[2] (Institute of Mathematics, Academy of Mathematics and Systems Science,
Chinese Academy of Sciences, China)

Wirsing M, Lu RQ. Preface: Foundations and practice of systems and software engineering — festschrift in honor of Prof. Manfred Broy. *Int J Software Informatics*, **Vol.5, No.1-2 (2011), Part I: 1–6. http://www.ijsi.org/1673-7288/5/i90.htm**

This Festschrift contains papers written by friends, students and colleagues to honor Manfred Broy in celebration of his 60th birthday. It presents results in the area of systems and software engineering - Manfred's research area - comprising both theoretical and practical aspects.

The Festschrift is divided into two volumes. The first volume contains ten scientific contributions and presents results for the more foundational issues of Manfred's research area ranging from formal verification and specification to concurrency and formal transformation. The second volume comprises seven papers and is devoted to more practical aspects of systems and software engineering. It presents interesting approaches to model-driven, automated, and change-driven software development and discusses issues in feature-oriented modeling, abstraction patterns and framework construction. The papers of both volumes have a clear relation with Manfred's research; some of them even compare or extend Manfred's work, and all papers contain personal notes of the authors where they explain there relationship with Manfred.

Manfred Broy is one of the internationally most well-known and most influential German computer scientists. He is a great researcher and teacher in the area of software and systems development where he always bases his methods on clear logical and mathematical principles and techniques. His aim is to master the specification and development of complex software systems by applying well-thought processes, designing flexible software architectures and using modern tools.

Manfred was born in 1949. He received in his Diploma degree in Mathematics and his PhD and Habilitation in Computer Science from Technical University of Munich in 1976, 1980, and 1982, respectively. His advisor was Friedrich L. Bauer,

Corresponding author: Martin Wirsing, Email: wirsing@lmu.de
Received 2011-05-12

the founding father of German Computer Science. Manfred went on as a researcher and scientific assistant at the Technical University of Munich until 1983, then as full professor at the University of Passau, Germany, from 1983 to 1989, including a period as founding dean of the Faculty of Mathematics and Computer Science from 1983-1987, then back as full professor at Technical University of Munich and chair of Systems and Software Engineering from 1989, which is his current position and where he was also the founding dean of the new faculty of Computer Science in 1992. Manfred is a member of the IFIP Working Groups 2.2 and 2.3, of the European Science Academy and the German Academy Leopoldina of Natural Scientists; he has received numerous further honors and prizes including a Doctor Honoris Causa from University of Passau and the most prestigious Gottfried Wilhelm Leibniz Prize of the German Science Foundation.

Manfred's more than 300 articles and papers show that his research interests span all core topics of software and systems engineering and include foundations, methods, processes, models, description techniques and tools. A main characteristic of his research is that all his contributions are always based on clear and sound mathematical and logical foundations. In the beginning of his career he was a member of F.L. Bauer's and Klaus Samelson's research group working in the CIP project on "Computer-Aided Intuition Guided Programming"[2]. The (at that time) revolutionary idea of the project was to consider programming as a fully formal activity[28] where program transformations are the key for a rigorous machine-supported top down approach to derive executable programs from formal requirement specifications. To achieve this a formal notion of program transformation, a new approach to algebraic specification of data types and the first wide spectrum language CIP-L[1] were developed by Manfred and his colleagues (among them MW – one of the editors of this volume). The CIP-L language was covering all phases of the program development process ranging from non-executable logical notations for specifying requirements and applicative programming concepts and abstract data types for software design to procedural and imperative programming constructs; program transformations were formally connecting the different levels of abstraction and in many cases correctness proofs were ensuring the soundness of the translations. 20 years later these ideas found a revival in model driven development and in particular in the Model-Driven Architecture MDA of OMG - with the difference that models and model transformations replace (wide spectrum) programs and program transformations (cf. Ref.[42]).

Manfred made important contributions to all aspects of the CIP approach. He was involved in the design and the formal mathematical semantics of CIP-L; he developed many program transformations (see e.g. Refs.[3, 25, 31]); in the area of algebraic specifications he was one of the authors of the so-called loose semantics[36], the theories of hierarchical[46] and of partial abstract data types[33], the algebraic semantics[37] of many programming language concepts (including nondeterminism[32], gotos[22], and communicating sequential processes[34]), and studied the relation between algebraic specification and program correctness[27]. But Manfred was not only interested in theory and semantics; he always found it very important to carry out case studies and to validate theoretical results in practice. A good example is his use of the Larch proof assistent for verifying program developments (see Ref.[12]); 20 years later, Tobias Nipkow reworks the Majority Vote Algorithm - one of Manfred's case studies

- in the Isabelle/HOL theorem prover (see Ref.[43] in this volume). Another very nice example is the following: in 1979 Albert Endres from the German IBM branch challenged Manfred by claiming that a fully formal description of a text editor would be infeasible. Of course, Manfred accepted this challenge and won it by writing a very clear formal specification in CIP-L (see Ref.[6]).

Since the beginning of the 80ties concurrency and interaction are the guiding topics of Manfred's research. Already in his PhD thesis[5] he studies key issues of concurrent programming like fairness, nondeterminism, and communication and uses program transformations for defining an operational and a declarative semantics of parallel and concurrent program constructs.

In his habilitation thesis (see also Ref.[7]) he develops a new completely formal "theory for nondeterminism, parallelism, communication and concurrency" where communicating systems are described by mutually recursive systems of nondeterministic equations for functions and streams.

In the following years he refines and completes this theory resulting in the elegant denotational FOCUS approach to the specification and development of interactive systems[30]. The basic concept is a notion of a finite or infinite stream of messages for describing the communication history between two entities of a data flow network; then a network component is represented by a stream-processing function mapping input streams to output streams, and a concurrent system is constructed by composing stream-processing functions using appropriate composition operators such as sequential composition, parallel composition, and feedback. In several further excellent papers Manfred equips the stream model with a compositional notion of refinement[13] and extends it to cover distributed[8], timed[11], and mobile systems[4] (for a comprehensive presentation see also Bernhard Rumpe's paper in this volume[44]).

As Manfred writes in Ref.[25], systems and software engineering is "organized in a development process for carrying out a series of modeling tasks in a systematic way. It requires modeling and description of the application domain, software requirements, software architecture, software components, their implementation and deployment". In contrast to the usual practice where most of these tasks rely on informal text and graphical description techniques Manfred's claim is that all these tasks should and can be "completely carried out within a mathematically and scientifically respectable theory"[30].

In his engineering approach a mathematical "system model" serves a basis for system specification, system development by refinement, and system implementation[15,18]. Manfred chooses the stream model as his system model and shows that many design notations such as state machines[35], Message Sequence Charts[14], SDL[9], and UML-RT components[40] can be scientifically founded on the stream model; refinement notions such as component glass box[18] and blackbox refinement can be defined on top of the refinement for stream-processing functions. In recent papers he gives concise mathematical foundations of UML[20] and of service-oriented computing[19,26] in terms of stream-processing functions (for an insightful comparison with a feature-oriented approach see the paper of Lengauer and Apel[41] in this volume). The system model provides also a basis for powerful tool support. His AutoFocus tool[39] supports component-based modeling of reactive, distributed systems and provides Java code generation, automatic test case generation and model checking of specifications and refinements.

Manfred is also very influential in practical issues of systems and software engineering. He is one of the proponents of the standard process model V-model RT[29] for German governmental software projects and is involved in discussions about the nature of components[21], challenges for software-intensive systems[16] and automotive software engineering[17]. Current topics of Manfred's research are requirements engineering[38], architectural models[24], quality assurance[45], and development methods[23] for complex industrial-scale software systems in the context of various industry cooperations in the telecommunications, avionics, automotive, banking, and business information systems domains. As a teacher he was shaping the computer science curriculum of the University of Passau in the eighties; he wrote four standard volumes of introduction to computer science[10], he created recently a new master degree on automotive software engineering and is involved in a Bavarian elite master degree on software engineering. Last not least since about 20 years he is director of the famous Marktoberdorf summer schools for young computer scientists and mathematicians working in the field of formal software and systems development.

The influence of Manfreds scientific contribution is not limited to the computer science community in Germany. Neither is it limited to the European or American region. One of the editors of this volume, RL, has witnessed how Chinese computer scientists, in particular their young generation, benefited a lot from Manfreds lessons and expertise. RL met him the first time in Enschede, Netherland, in 1978 during an IFIP Working Group 2.4 meeting and then in Munich at a meeting of F.L. Bauer's group. Since then Manfred has taken care of sending regularly the TUM technical reports in computer science to the Institute of Mathematics of the Chinese Academy of Sciences. After his moving to the University of Passau, the Chinese Academy of Sciences again received research reports from Passau for many years. All these reports have helped many Chinese computer scientists in getting access to the research frontiers in programming theory and formal methods of software engineering at that time. Together with Bernd Krieg-Brückner, formerly also a member of F.L. Bauer's group and currently a leading professor at FB3 Bremen University, Manfred visited China in 1985 to give a series of lectures on the programming language Ada, abstract data types and program transformation. Young researchers and students from many Chinese universities and research institutions attended these lectures. Many of them remember this valuable experience even after several decades.

As editors of the Festschrift we thank all authors for their scientific contributions. We also would like to take the occasion to wish Manfred many more scientifically fruitful years to come.

Martin Wirsing and Ruqian Lu

References

[1] Bauer FL, Berghammer R, Broy M, Dosch W, Geiselbrechtinger F, Gnatz R, Hangel E, Hesse W, Krieg-Brückner B, Laut A, Matzner T, Möller B, Nickl F, Partsch H, Pepper P, Samelson K, Wirsing M, Wössner H. The Munich Project CIP, Volume I: The Wide Spectrum Language CIP-L. LNCS 183, Springer, 1985.

[2] Bauer FL, Broy M, Dosch W, Gnatz R, Krieg-Brückner B, Laut A, Luckmann M, Matzner T, Möller B, Partsch H, Pepper P, Samelson K, Steinbrüggen R, Wirsing M, Wössner H. Programming in a wide spectrum language: A collection of examples. Sci. Comput. Program., 1981, 1(1-2):73–114.

[3] Bauer FL, Broy M, Partsch H, Pepper P, Wössner H. Systematics of transformation rules. In Bauer FL, Broy M, eds. Program Construction. LNCS 69, Springer, 1978. 273–289.

[4] Bergner K, Grosu R, Rausch A, Schmidt A, Scholz P, Broy M. Focusing on mobility. In: Ralph H. Sprague Jr., eds. Proc. of the Thirty-Second Annual Hawaii International Conference on System Sciences. IEEE Computer Society, 1999.

[5] Broy M. Transformation Parallel Ablaufender Programme. Dissertation, Technische Universität München, München, 1980.

[6] Broy M. Zur Spezifikation von Programmen für die Textverarbeitung. In: Wossidlo PR, ed. Textverarbeitung und Informatik. volume 30 of Informatik-Fachberichte, Springer, 1980. 75–93.

[7] Broy M. A theory for nondeterminism, parallelism, communication, and concurrency. Theor. Comput. Sci., September 1986, 45: 1–61.

[8] Broy M. Requirement and design specification for distributed systems. In Friedrich H. Vogt, editor, Concurrency, volume 335 of Lecture Notes in Computer Science, Springer, 1988. 33–62.

[9] Broy M. Towards a formal foundation of the specification and description language SDL. Formal Asp. Comput., 1991, 3(1): 21–57.

[10] Broy M. Informatik - eine grundlegende Einführung. Teil I-IV. Springer Verlag, 1992.

[11] Broy M. Functional specification of time-sensitive communicating systems. ACM Trans. Softw. Eng. Methodol., 1993, 2(1): 1–46.

[12] Broy M. Experiences with software specification and verification using lp, the larch proof assistant. Formal Methods in System Design, 1996, 8(3): 221–272.

[13] Broy M. Compositional refinement of interactive systems. J. ACM, November 1997, 44: 850–891.

[14] Broy M. On the meaning of message sequence charts. In: Lahav Y, Wolisz A, Fischer J, Holz E, eds. 1st Workshop of the SDL Forum Society Workshop on SDL & MSC. 1998. 13–34.

[15] Broy M. Toward a mathematical foundation of software engineering methods. IEEE Trans. Softw. Eng., January 2001, 27: 42–57.

[16] Broy M. The 'Grand Challenge' in Informatics: Engineering software intensive systems. IEEE Computer, 2006, 39(10): 72–80.

[17] Broy M. Challenges in automotive software engineering. In: Osterweil LJ, Rombach HD, Soffa ML, eds. ICSE. ACM, 2006. 33–42.

[18] Broy M. A logical basis for component-oriented software and systems engineering. Comput. J., 2010, 53(10): 1758–1782.

[19] Broy M. Multi functional software systems: Structured modeling and specification of functional requirements. Sci. Comput. Program., 2010, 75(12): 1193–1214.

[20] Broy M, Cengarle MV, Rumpe B. Semantics of UML, Towards a System Model for UML, Part 1 and 2. Technical report, Technische Universität München, 2007.

[21] Broy M, Deimel A, Henn J, Koskimies K, Plasil F, Pomberger G, Pree W, Stal M, Szyperski CA. What characterizes a (software) component? Software - Concepts and Tools, 1998, 19(1): 49–56.

[22] Broy M, Dosch W, Möller B, Wirsing M. GOTOs - a study in the algebraic specification of programming languages (extended abstract). In: Brauer W, ed. GI Jahrestagung, volume 50 of Informatik-Fachberichte, Springer, 1981. 109–121.

[23] Broy M, Feilkas M, Herrmannsdoerfer M, Merenda S, Ratiu D. Seamless model-based development: from isolated tools to integrated model engineering environments. Proc. of the IEEE. April 2010, 98(4): 526–545. Available at http://dx.doi.org/10.1109/JPROC.2009.2037771.

[24] Broy M, Gleirscher M, Kluge P, Krenzer W, Merenda S, Wild D. Automotive architecture framework: towards a holistic and standardised system architecture description. Technical Report TUMI0915, Technische Universität München, 2009.

[25] Broy M, Krieg-Brückner B. Derivation of invariant assertions during program development by

transformation. ACM Trans. Program. Lang. Syst., 1980, 2(3): 321–337.
[26] Broy M, Krüger IH, Meisinger M. A formal model of services. ACM Trans. Softw. Eng. Methodol., 2007, 16(1).
[27] Broy M, Möller B, Pepper P, Wirsing M. Algebraic implementations preserve program correctness. Sci. Comput. Program., 1986, 7(1): 35–53.
[28] Broy M, Pepper P. Program development as a formal activity. IEEE Trans. Software Eng., 1981, 7(1): 14–23.
[29] Broy M, Rausch A. Das neue V-Modell XT. Informatik Spektrum. 2005, 28(3): 220–229.
[30] Broy M, Stølen K. Specification and development of interactive systems: focus on streams, interfaces, and refinement. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.
[31] Broy M, Wirsing M. Program development: from enumeration to backtracking. Inf. Proc. Lett., 1980, 10(4/5): 193–197.
[32] Broy M, Wirsing M. On the algebraic specification of nondeterministic programming languages. In: Egidio Astesiano and Corrado Böhm, eds. CAAP. LNCS 112, Springer, 1981. 162–179.
[33] Broy M, Wirsing M. Partial abstract types. Acta Inf., 1982, 18: 47–64.
[34] Broy M, Wirsing M. On the algebraic specification of finitary infinite communicating sequential processes. In: Björner D, ed. IFIP TC2 Working Conference on Formal Description of Programming Concepts II, pages 171–196. North Holland Publ., Company, 1983.
[35] Broy M, Wirsing M. Invited talk: Algebraic State Machines. In: Rus T, ed. AMAST. LNCS 1816, Springer, 2000. 89–188.
[36] Broy M, Wirsing M, Pair C. A systematic study of models of abstract data types. Theor. Comput. Sci., 1984, 33: 139–174.
[37] Broy M, Wirsing M, Pepper P. On the algebraic definition of programming languages. ACM Trans. Program. Lang. Syst., 1987, 9(1): 54–99.
[38] Méndez Fernández D, Penzenstadler B, Kuhrmann M, Broy M. A meta model for artefact-orientation: Fundamentals and lessons learned in requirements engineering. In: Petriu DC, Rouquette N, Haugen O, eds. MoDELS (2), LNCS 6395, Springer, 2010. 183–197.
[39] Hölzl F, Feilkas M. AutoFocus 3: a scientific tool prototype for model-based development of component-based, reactive, distributed systems. Proc. of the 2007 International Dagstuhl Conference on Model-based Engineering of Embedded Real-Time Systems, MBEERTS'07. Berlin, Heidelberg, Springer-Verlag, 2010. 317–322.
[40] Krüger I, Prenninger W, Sandner R, Broy M. From scenarios to hierarchical broadcasting software architectures using UML-RT. International Journal of Software Engineering and Knowledge Engineering, 2002, 12(2): 155–174.
[41] Lengauer C, Apel S. Feature-oriented system design and engineering. Int. J. Software and Informatics, 2011, 5(1-2): 231–244.
[42] Mellor SJ, Clark AN, Futagami T. Guest editors' introduction: Model-driven development. IEEE Software, 2003, 20(5): 14–18.
[43] Nipkow T. Majority Vote Algorithm revisited again. Int. J. Software and Informatics, 2011, 5(1-2): 21–28.
[44] Ringert JO, Rumpe B. A little synopsis on streams, stream processing functions, and state-based stream processing. Int. J. Software and Informatics, 2011, 5(1-2): 29–53.
[45] Wagner S, Broy M, Deißenböck F, Kläs M, Liggesmeyer P, Münch J, Streit J. Softwarequalitätsmodelle - Praxisempfehlungen und Forschungsagenda. Informatik Spektrum, 2010, 33(1): 37–44.
[46] Wirsing M, Pepper P, Partsch H, Dosch W, Broy M. On hierarchies of abstract data types. Acta Inf., 1983, 20: 1–33.