Dr.-Ing. Robert Laurence Baber
Professor
Department of Computing and Software
McMaster University
2003 January 13

Robert L. Baber

Dr.-Ing. Robert Laurence Baber

Contents

Personal information 1

Education 2

Overview of professional experience 3

Current and recent university experience 5

Professional status, service and activities 7

Descriptions of earlier professional experience 10

Descriptions of experience concurrent with education 12

References 13

Publications 15

Recent lectures 18

Research interests 21

Dr.-Ing. Robert Laurence Baber

Personal information

Born:	1937 December 11 in Los Angeles, California, U.S.A.
Family:	married since 1964 September 4 to Ursula nee Kettler. Two children: Ingrid Anne, born 1966 June 19; Eric Robert, born 1969 August 21
Citizenship:	Federal Republic of Germany
Address:	82 Oneida Blvd. Ancaster, ON, L9G 4S5 Canada
Electronic mail:	home: Bob@RLBaber.com office: BobUniv@RLBaber.com, Baber@McMaster.CA
Web pages:	http://www.cas.mcmaster.ca/~baber, http://www.RLBaber.com
Tel. + Fax: Cell tel.:	+1-905-304-8099 (home) +1-905-979-3210
Languages:	fluent in English and German elementary ability in Esperanto limited passive ability in Swedish and Spanish
Hobbies:	opera, music, travelling, digital photography, ancient technology, stamp collecting Earlier: sailing, amateur theater, scuba diving, amateur radio

Dr.-Ing. Robert Laurence Baber

Education

Schooling: 1943 September to 1954 June in Modesto, California; Seattle, Washington; Anchorage, Alaska; San Antonio, Texas; Alamo Heights, Texas; Roswell, New Mexico

University: 1954 September — 1962 January
Massachusetts Institute of Technology, Cambridge, Massachusetts, U.S.A.

Degrees: Bachelor of Science in Electrical Engineering, 1959 June 12
Master of Science in Electrical Engineering, 1959 June 12
Master of Science in Industrial Management, 1962 January 30

Doktor-Ingenieur (Doctor of Engineering) in Informatik (computing science), Technische Hochschule Darmstadt (now Technical University of Darmstadt), Germany, 1994 September 30. My dissertation was on the practical applicability of mathematically rigorous methods for ensuring the correctness of sequential computer programs. Prof. Dr.-Ing. Hans-Jürgen Hoffmann was my doctoral advisor; Prof. Dr. David L. Parnas was co-assessor.

Honors: elected to honor societies: Tau Beta Pi (engineering), Eta Kappa Nu (electrical engineering) and Sigma Xi (scientific research)

Additional studies: 1959 September — 1960 June, nuclear engineering (M.I.T.)

1975 October — 1979 February, mathematics (probability theory), part time, Johann Wolfgang Goethe University, Frankfurt/Main, Germany

Dr.-Ing. Robert Laurence Baber

Overview of professional experience

1956 Sept. —
1958 Sept.: While an undergraduate student at M.I.T., I was selected for the cooperative course in electrical engineering, a program leading to the simultaneous award of the Bachelor and Master of Science degrees. This program included four terms of industrial experience (instead of summer vacations), during which time I was employed in laboratories of the IBM Corporation.

1959 July —
1959 Sept.: Valley National Bank, Phoenix, Arizona. Summer job after graduation in 1959 June.

1961 Sept. —
1962 Feb.: Department of Industrial Management, M.I.T. Part time research assistantship helping to formulate and programming a tax research model.

1962 Jan. —
1962 Feb.: Prof. Martin Greenberger (professor at M.I.T. and consultant). Assisting in formulating and programming a model for a research project on television viewing behaviour.

1962 March —
1964 March: U.S. Army Signal Corps, Ft. Gordon, Georgia; Washington, D.C. and Ft. Ritchie, Maryland. Military service as a commissioned officer. Programming, systems design and maintenance of externally supplied command and control software systems.

1964 March —
1966 March: Control Data Corporation, Minneapolis, Minnesota, U.S.A. and Frankfurt/Main, Germany. Technical sales support.

1966 March —
1970 March: Diebold Europe S.A., Frankfurt/Main, Germany. Consultant and Senior Consultant. EDP consulting.

1970 April —
1975 March: Harbridge House Europe, Frankfurt/Main, Germany. Senior Consultant. Management consulting.

1975 April —
1996 Jan.: Self-employed independent software engineering and management consultant, external lecturer (“Lehrbeauftragter”) at the Johann Wolfgang Goethe University, Frankfurt/Main, Germany.

1996 Feb. — 1999 Dec.: Department of Computer Science, University of the Witwatersrand, Johannesburg, South Africa. Visiting professor.

2000 Jan. — present: Department of Computing and Software, McMaster University, Hamilton, Ontario, Canada. Professor.

Dr.-Ing. Robert Laurence Baber

Current and recent university experience

2000 Jan. — present: Professor, Department of Computing and Software, McMaster University, Hamilton, Ontario, Canada. Each year at McMaster I have taught the following four courses. The abbreviations SE1, etc. indicate the year of study in the software engineering programme in which the topic was taught.

- Software Design II (SE2)
- Software and Social Responsibility (SE3)
- Design of Human-Computer Interfaces (SE4)
- Mathematically Rigorous Software Design (SE and CS, 4 and Masters)

I continued to revise and add new material to my text book for the course Mathematically Rigorous Software Design. This latest version of this material is currently available publicly on the internet at http://www.cas.mcmaster.ca/~baber/Courses/46L03/MRSDLect.pdf.

I was the first instructor to teach the latter three courses listed above in the new Software Engineering programme at McMaster. Based on given goals and general statements of the desired content, I designed the specific contents of the courses Software and Social Responsibility and Design of Human-Computer Interfaces. The fourth course, Mathematically Rigorous Software Design, was essentially the same course that I taught earlier at other universities. At McMaster I continued to revise and develop new material relevant for this course and a possible follow-on course.

At McMaster I am co-superving three master's students. I have served on examination committees of several master's and doctorate students. I also served as an external examiner for a thesis prepared by a doctoral student at the University of South Africa (UNISA).

Recently I wrote a short text book for software engineering students entitled *Translating English to Mathematics*. I have published this draft on the internet and plan to revise it from time to time.

I have also served on various administrative committees at McMaster University.

1996 Feb. — 1999 Dec.: Visiting professor, Department of Computer Science, University of the Witwatersrand, Johannesburg, South Africa. At Wits I taught the following topics. The abbreviations CS1, etc. indicate the year of study in which the topic was taught.

1996:

- Mathematically Rigorous Software Development (MRSD), CS Hons
- MRSD, CS Masters
- Limits of Computing (LoC), CS1
- Software Engineering (SE), CS3

1997:
- MRSD, CS Hons
- Formal Aspects of Computing (FAC), CS3
- LoC, CS1

1998:
- MRSD, CS Hons
- Formal Languages and Automata (FLA, renaming of FAC above), CS3
- Data and Data Structures (DDS), CS1
- LoC, CS1 (jointly with another instructor)

1999:
- MRSD, CS Hons
- Formal Languages and Automata, CS3
- Data and Data Structures (DDS), CS1

For the topics MRSD I significantly revised and translated into English my earlier lecture notes and exercises. While I was at Wits this new material was available publicly on internet.

In 1998 I revised the topic "Data and Data Structures" and have written a new set of lecture notes for it.

At the request of Dr. Robert L. Glass I wrote the chapter "The prolonged metamorphosis of a software *engineer*" for a book he compiled and edited entitled *In the Beginning — Personal Recollections of Software Pioneers*.

Administrative assignments: 1997: Honours course coordinator, Library and book coordinator for the Computer Science Department

mid 1997 - present: Chairman of the Maths-Sciences Library Committee

Dr.-Ing. Robert Laurence Baber

Professional status, service and activities

Professional: Professional Engineer (P.Eng., Ontario professional engineering registration)
Chartered Engineer (CEng, U.K. professional engineering registration)
European Engineer (Eur. Ing., European engineering registration)
Fellow of the British Computer Society (FBCS)

cofounder and first chairman of the Computer Chapter of the German Section of the IEEE (1986-1990)

Vice Chairman of the German Section of the IEEE (1991-1996)

Memberships: British Computer Society (BCS), Fellow
Institute of Electrical and Electronics Engineers (IEEE), Senior Member
IEEE Computer Society, IEEE Reliability Society
Gesellschaft für Informatik (GI, German Informatics Society)
Verein Deutscher Ingenieure (VDI, Society of German Engineers)
Verband Deutscher Elektrotechniker (VDE, Association of German Electrical Engineers)
South African Institute of Computer Scientists & Information Technologists (SAICSIT)
Association for Computing Machinery (ACM)

Who's Who listings: *International Who's Who in Engineering*, *Men of Achievement*, *International Who's Who of Intellectuals* and other biographical reference works

Referee: Refereed articles for the journals *IEEE Computer*, *IEEE Potentials*, *High Integrity Systems*, *Advances in Engineering Software* and the *South African Computer Journal* and for the conferences *SAICSIT 2000*, *SAICSIT 2001*, *SAICSIT 2002*, *Workshop on Inspection in Software Engineering 2001*, *Software Quality Engineering 1997* and *IFIP International Workshop on Dependable Computing and Its Applications (DCIA) 1998*. Reviewed/refereed several books for John Wiley, Inc.

Conference Committees: Member of the Program Committees for *SAICSIT 2000*, *SAICSIT 2001* and *SAICSIT 2002*, Member of the International Scientific Advisory Committees for the conferences *Software Quality Management 1993*, *1994* and *1995* and *Software Quality Engineering 1997*, member of the Programme Committee of *IFIP International Workshop on Dependable Computing and Its Applications (DCIA) 1998* and Southern African Liaison for *The Ninth International Symposium on Software Reliability Engineering ISSRE 1998*.

Books: *Software Reflected: The Socially Responsible Programming of Our Computers*, North-Holland 1982, German translation: Springer-Verlag 1986, Polish translation: WNT 1989

The Spine of Software: Designing Provably Correct Software — Theory and Practice, Wiley 1987

Error-free Software: Know-how and Know-why of Program Correctness, English translation: Wiley 1991, German original: Oldenbourg 1990, Russian translation: Nauka, in preparation

Praktische Anwendbarkeit mathematisch rigoroser Methoden zum Sicherstellen der Programmkorrektheit, Walter de Gruyter, Berlin, 1995.

The VDI (Verein Deutscher Ingenieure) technical working group 4.1 on software reliability compiled a book (*Software-Zuverlässigkeit*, VDI Verlag, 1993) on this subject to which I contributed a section on applying correctness proof techniques in practice.

Mathematically Rigorous Software Design, a text book for one of my university courses, published personally on the internet and revised from time to time.

Translating English to Mathematics, draft of a short text book for students, published personally on the internet and to be revised from time to time.

Seminar: In 1985 I conceived, planned and developed a three day seminar on designing provably correct software. I have instructed this seminar 22 times, both in English and in German, in Germany, the Netherlands, Denmark, Tunisia (for a German software house), the U.K. and, upon the invitation of the Institute of Software of the Chinese Academy of Sciences (Academia Sinica), in Beijing, China.

University teaching: As an external lecturer ("Lehrbeauftragter") of the Informatics (Computing Science) Department of the Johann Wolfgang Goethe University, Frankfurt/Main, Germany, I developed and taught a course entitled "Einführung in die Konstruktion fehlerfreier Software für praktische Anwendungen" (Introduction to the design of error free software for practical applications). This course was largely based on my book, *The Spine of Software*. I also designed and taught a follow-on seminar on the same subject entitled "Projektstudien zur Konstruktion fehlerfreier Software" (Project studies on designing error free software). I taught these courses from the fall of 1988 until early 1996.

Publications and lectures: I have delivered many invited lectures and have written a number of professional papers, including "Epilogue: Future Developments", the concluding chapter in the *Software Engineer's Reference Book* (Butterworth Scientific Ltd., Guildford, U.K., 1991), and "Proofs of Correctness", a section in the *Encyclopedia of Software Engineering* (John Wiley & Sons Inc., New York, 1994). See list of publications below.

Technical activities: While in Germany I participated in the following technical working groups:
VDI Working Group 4.1 on Software Reliability
VDI Technical Committee 4.4 on Safety of Automatic Control Systems
GI Working Group 7.4 on professional personnel
GI-CEPIS German working group on EISS (European Informatics Skills Structure)

Other: For a number of years prior to 1996 February I interviewed applicants for admission to the Massachusetts Institute of Technology in Cambridge, Massachusetts, U.S.A. These applicants were in their last year of high school and included dependents of U.S. military personnel stationed in Germany, German Gymnasium students and foreign residents of Germany.

Dr.-Ing. Robert Laurence Baber

Descriptions of earlier professional experience

1962 March —
1964 March: U.S. Army Signal Corps, Ft. Gordon, Georgia; Washington, D.C. and Ft. Ritchie, Maryland. Military service as a commissioned officer. Programming, systems design and software maintenance. This work involved a variety of tasks ranging from designing and writing small ancillary programs, library macros, utility programs, a program library system, etc., to debugging and modifying a large command and control system supplied by a software house. I also revised and reprogrammed the procedures for converting original input data into the form and sequences required by a contractor-supplied application software system. This work involved programming the IBM 1401 and CDC 1604 computers in assembly language, Fortran and Jovial.

1964 March —
1966 March: Control Data Corporation, Minneapolis, Minnesota, U.S.A. and Frankfurt/Main, Germany. My assignments in the marketing organization of Control Data Europe dealt with general technical sales support and instructing various courses in computers and applications to both customers and CDC personnel. This work involved advising salesmen and customers regarding system configuration, helping to prepare proposals for computer systems, assisting programmers with various problems and questions, designing and teaching courses ranging from introductory ones for non-technical people to relatively advanced systems programming techniques for computer specialists with substantial experience, etc. During this time I used the CDC 3200, 3400 and 3600 computer systems and assembly language, Fortran and Algol.

1966 March —
1970 March: Diebold Europe S.A., Frankfurt/Main, Germany. Consultant and Senior Consultant. My position with Diebold Europe involved a mixture of EDP consulting, instructing parts of seminars and performing research and preparing reports, conference presentations, etc., within the scope of the Diebold Research Program—Europe, a cooperative program sponsored by a number of European business and governmental organizations. This program dealt with many different aspects of the application of computer systems to management and business problems. The main objectives were to identify major opportunities and problems for sponsoring organizations arising from developments in information technology and to exchange knowledge and experience gained in these areas. Topics of reports, conference presentations, etc., included developments in hardware and software technology, common data base, integrated management information systems, planning and implementing information systems, EDP personnel, management of the EDP function, etc.

1970 April —
1975 March: Harbridge House Europe, Frankfurt/Main, Germany. Senior Consultant. While with this management consultancy, I participated in and directed consulting and seminar projects on a wide range of business topics such as management, mar-

keting, sales, systems and procedures, inventory control, logistics, etc. Many of these assignments involved conceiving, planning, designing, developing and assessing computer based information systems.

1975 April — 1996 Jan.: Self-employed independent software engineering and management consultant. My consulting, software design and seminar projects covered a range of topics similar to those on which I worked while employed by Harbridge House (1970 April to 1975 March), but generally involved EDP topics, the planning, design and development of application software systems, etc., to a greater degree. These consulting assignments included the design and development of software systems for business and technical applications and written in various programming languages for execution on computer systems of various sizes, types and makes.

During this time I also

- taught courses in the Computing Science Department (Fachbereich Informatik) at the Johann Wolfgang Goethe University, Frankfurt/Main, Germany, as an external lecturer,
- participated in several technical working groups of German professional societies,
- developed and instructed a seminar for software developers in industrial and commercial practice,
- wrote four books,
- wrote a number of professional papers,
- gave a number of invited lectures at conferences and workshops and
- refereed papers for professional journals.

See other sections for more details on these activities.

Dr.-Ing. Robert Laurence Baber

Descriptions of experience concurrent with education

Within the framework of the cooperative course in electrical engineering at M.I.T.:

1956 Sept. —
1957 Jan.: IBM Product Development Laboratory, Poughkeepsie, N.Y. Design of transistor circuits for the IBM 608 calculator, the firm's first fully transistorized machine to go into commercial production.

1957 June —
1957 Sept.: IBM Time Equipment Division, Endicott, N.Y. Preliminary design of electromechanical calculating circuitry for the IBM Automatic Production Recorder System and of an interface between this machine and a stored program computer.

1958 Feb. —
1958 Sept.: Design Automation Group, IBM Product Development Laboratory, Poughkeepsie, N.Y. Programming sections of a system for checking computer logic diagrams for correctness and conformity with certain design rules. I designed and wrote a program generator to create most of these routines. The program generator and the generated routines were written in the IBM 705 Autocoder language (a macro-assembly language).

Summer job after graduation in 1959 June:

1959 July —
1959 Sept.: Valley National Bank, Phoenix, Arizona. Investigated possibilities of improving bank business procedures, designed and instructed an introductory course in EAM (electrical accounting machine) principles for bank employees.

Dr.-Ing. Robert Laurence Baber

References

Prof. Dr.-Ing. habil. Fevzi Belli
Raum: P 5207, Fachbereich 14 Elektrotechnik
Universität-GH Paderborn
Warburger Straße 100
D-33098 Paderborn, Germany
Tel.: +49-5251-60-3448/3447
Fax: +49-5251-60-3246
Email: belli:@adt.uni-paderborn.de

Prof. Dr. Timm Grams
Fachbereich Elektrotechnik
Fachhochschule Fulda
Marquardstraße 35
D-36039 Fulda, Germany
Tel.: +49-661-9640-556
Fax: +49-661-9640-559
Email: Timm.Grams@et.fh-fulda.de

Prof. Dr.-Ing. Hans-Jürgen Hoffmann
Fachbereich Informatik, FG PU
Technische Universität Darmstadt
Wilhelminenstr. 7, S4/03/C122
D-64283 Darmstadt, Germany
Tel.: +49-6151-163410
Fax: +49-6151-166229
Email: hjhoffmann@informatik.tu-darmstadt.de *(not for requesting personal assessments)*

Dr. Philip Machanick
School of Information Technology and Electrical Engineering
The University of Queensland
St. Lucia, QLD 4072 Australia
Tel.: +61-7-336-52766
Email: philip@itee.uq.edu.au

Dr. Conrad Mueller
Head of School
School of Computer Science
University of the Witwatersrand, Johannesburg
Private Bag 3
2050 Wits, South Africa
Tel.: +27-11-716-3344
Fax: +27-11-339-3513
Email: conrad@cs.wits.ac.za

Prof. Dr. David Lorge Parnas
Director of the Software Quality Research Laboratory
SFI Fellow, Professor of Software Engineering
Department of Computer Science and Information Systems
Faculty of Informatics and Electronics
University of Limerick
Limerick, Ireland
Tel.: +353-61-202731
Fax: +353-61-202734
Email: David.Parnas@UL.ie

Dr. Paul A. Taylor
Chair, Department of Computing and Software
McMaster University
1280 Main Street West
Hamilton, ON, L8S 4K1, Canada
Tel.: +1- 905-525-9140 ext. 24952
Fax: +1-905-524-0340
Email: taylorpa@mcmaster.ca

Prof. Dr. hab. Wladyslaw M. Turski
Institute of Informatics
Warsaw University
ul. Banacha 2
02-097 Warsaw, Poland
Tel: +48-2-658-3522
Fax: +48-2-658-3164
Email: wmt@mimuw.edu.pl *(not for requesting personal assessments)*

Dr. André van der Poll
Department of Computer Science and Information Systems
University of South Africa
PO Box 392
Unisa, 0003, South Africa
Tel. +27-12-429-6230
Email: vdpolja@unisa.ac.za

Dr.-Ing. Robert Laurence Baber

Publications

Translating English to Mathematics, short monograph/text book published on the internet at http://www.cas.mcmaster.ca/~baber/Courses/General/EnglToMath.pdf, 2002.

Mathematically Rigorous Software Design, text book for students and interested others published on the internet at http://www.cas.mcmaster.ca/~baber/Courses/46L03/MRSDLect.pdf, revised and new material continually added 2000-2002.

"Software Engineering Education: Issues and Alternatives", *Annals of Software Engineering*, Vol. 6 (special volume on "Software Engineering Education"), pp. 39-59, 1998 (1999 April).

"On the Inadequacy of Ordinary Preconditions for the Practical Design and Verification of Programs", in *Proceedings The Ninth International Symposium on Software Reliability Engineering, Paderborn, Germany, November 4-7, 1998*, pp. 43-48, IEEE Computer Society, Los Alamitos, California.

"Comparison of electrical 'engineering' of Heaviside's times and software 'engineering' of our times", *IEEE Annals of the History of Computing*, Vol. 19, No. 4, pp. 5-17, 1997 Oct.-Dec.

"The prolonged metamorphosis of a software *engineer*", in *In the Beginning — Personal Recollections of Software Pioneers*, Glass, Robert L. (ed.), IEEE Computer Society Press, 1997.

"CS Education and an Engineering Approach to Software Development", in *Proceedings of the 27th Southern African Computer Lecturer's Association Conference*, 1997 June 22-24, 97/10, Publication series, Department of Computer Science and Information Systems, University of Port Elizabeth, Port Elizabeth, South Africa, 1997 June, p. 20-28.

"The Ariane 5 explosion as seen by a software *engineer*", Internal Report number 1997-02, Computer Science Department, University of the Witwatersrand, Johannesburg, South Africa, http://www.cs.wits.ac.za/~bob/ariane5.htm, 1997 February.

"Mathematically Rigorous Software Development", course lecture notes publicly available at ftp://ftp.cs.wits.ac.za/pub/courses/hons, 1996 October.

Praktische Anwendbarkeit mathematisch rigoroser Methoden zum Sicherstellen der Programmkorrektheit, (book), Walter de Gruyter, Berlin, 1995.

"The Engineering of Engineering Software", *Boundary Elements Communications*, Vol. 5, No. 1, 1994 January, p. 6-11.

"Proofs of Correctness", in *Encyclopedia of Software Engineering*, John Wiley & Sons, New York, 1994, p. 925-930.

"Konzepte für die Erstellung möglichst fehlerfreier Software in der Vergangenheit und Zukunft", *HMD — Theorie und Praxis der Wirtschaftsinformatik*, Heft 163, 1992 Januar, S. 3-16.

Error Free Software: Know-How and Know-Why of Program Correctness, (own translation of the book *Fehlerfreie Programmierung für den Software-Zauberlehrling*, see below), John Wiley & Sons, Chichester, 1991.

"Portrait of a (Software) Engineer", *Journal of Systems and Software*, Vol. 15, 1991, p. 91-100.

"Epilogue: Future Developments", concluding chapter in the *Software Engineer's Reference Book*, Butterworth Scientific Ltd., Guildford, 1991.

"Software Development Tomorrow: Possible Future Worlds", *The Software Practitioner*, Vol. 1, No. 3-4, 1991 May, p. 12-13.

"Neue Ideen und Konzepte für die Erstellung nachprüfbar fehlerfreier Software", in *Entwicklung und Prüfung sicherheitsbezogener Systeme: Software- und Systemaspekte*, Meckelburg, Hans-Jürgen und Jansen, Herbert (eds.), VDE-Verlag, Berlin, Offenbach, 1990.

Fehlerfreie Programmierung für den Software-Zauberlehrling, (Buch), R. Oldenbourg Verlag, München, 1990.

"'Software engineering' vs. software *engineering*", *IEEE Computer* (Open Channel column), Vol. 22, No. 5, 1989 May, p. 81. Relevant letter to the editor and author's reply in *IEEE Computer*, Vol. 22, No. 9, 1989 September, p. 8.

"Software + Wartung = Widerspruch oder beliebte, bequeme Mythen entblößt", in *Softwarewartung* (Band 2 in der Reihe Angewandte Informatik), herausgegeben von B. Wix und H. Balzert, BI-Wissenschaftsverlag, Mannheim, Wien und Zürich, 1988, S. 105-122.

"Verläßlichkeit und Fehlertoleranz aus der Sicht eines Software-Ingenieurs", *Informationstechnik **it***, 30. Jahrgang 1988, Heft 3, S. 209-218.

The Spine of Software: Designing Provably Correct Software — Theory and Practice, (book), John Wiley & Sons, Chichester, 1987.

"User Interface: Window to an Informational World", in *User Interfaces: Gateway or Bottleneck?*, Thomas Bernold (ed.), Proceedings of the Technology Assessment and Management Conference of the Gottlieb Duttweiler Institute, User Interfaces International Conference, Rüschlikon, Zürich, Switzerland, 1986 October 20-21.

"Summary and Perspectives: Windows Revisited", in *User Interfaces: Gateway or Bottleneck?*, Thomas Bernold (ed.), Proceedings of the Technology Assessment and Management Conference of the Gottlieb Duttweiler Institute, User Interfaces International Conference, Rüschlikon, Zürich, Switzerland, 1986 October 20-21.

"Programmeren: Wetenschap of Gefröbel?", (Dutch translation by editor) *Informatie Maandblad voor Gegevensverwerking*, Vol. 28, No. 1, 1986 January, p. 34-40.

"I/O statements in higher programming languages: unnecessary and undesirable", *IEEE Computer* (Open Channel column), Vol. 18, No. 6, 1985 June, p. 112.

"Softwareentwicklung gestern, heute und morgen — Wissenschaft oder Flickwerk?", Berichte der Arbeitsgruppe Mathematisierung, Interdisziplinäre Arbeitsgruppe Mathematisierung (IAGM), Gesamthochschule Kassel, Heft 5, Februar 1985.

"Software Development: Science, Craft or Racket?", *Data Processing*, Vol. 26, No. 10, December 1984, p. 7-10.

"Ein Überblick über informationstechnologische Entwicklungen bis 1995", 6. Frühjahrstagung der Online-Benutzergruppe der DGD in Neu-Isenburg vom 8. bis 10. Mai 1984, Vorträge, DGD-Schrift (OLBG-5) 2/84, Frankfurt/Main, 1984.

"Software Development: Science or Patchwork?", *CWI Newsletter*, No. 2, March 1984, p. 18-34, published by the Centre for Mathematics and Computer Science, Amsterdam.

"De ontwikkeling van programmatuur gisteren, vandaag en morgen: Wetenschap of broddelwerk?", (Dutch translation by T. E. Timman), *I & I Kwartaalreeks over Informatie en Informatiebeleid*, No. 4, 1983.

"Der Kommentar: Informatikausbildung in der Bundesrepublik Deutschland: Der zukünftige Preis des gegenwärtigen Nichthandelns", *Informatik-Spektrum*, Band 6, Heft 1, Februar 1983, S. 36.

Software Reflected: the Socially Responsible Programming of Our Computers, (book), North-Holland Publishing Co., Amsterdam, 1982.

"A method for representing data items of unlimited length in a computer memory", *IEEE Transactions on Software Engineering*, Vol. SE-7, No. 6, Nov. 1981, p. 590-593.

"Software Reflected: the Land of Moc", *Systems, Objectives, Solutions (SOS)*, Vol. 1, No. 3, Aug. 1981, p. 105-117, published by the North-Holland Publishing Co., Amsterdam.

"Lagerwirtschaftsystem mit EDV: Gleichzeitige Optimierung von Wiederbestellmenge und Sicherheitsbestand", *Industrielle Organisation*, 42 (1973) Nr.9, Sept. 1973, S. 411-416.

"Wenn ein Informationssystem Selbstzweck wird", *Blick durch die Wirtschaft*, Frankfurt/Main, 3. Jan. 1972, S. 5.

"Data Compression Techniques", Diebold Research Program—Europe Professional Paper, Oct. 1968.

"The Integrated Management Information System", 3. Internationales Symposium Datenverarbeitung, Leipzig, März 1967.

"Information Theory and Statistical Analysis", Diebold European Research Program Professional Paper Series, 1966 Aug.

"Tape Searching Techniques", *Journal of the ACM*, Vol. 10, No. 4, Oct. 1963, p. 478-486.

"Computer Analysis and Evaluation of Stock Trading Tactics", Masters degree thesis, Massachusetts Institute of Technology, 1959 June.

Dr.-Ing. Robert Laurence Baber

Recent lectures

"Specification Driven Software Design", lecture given at the University of Westminster, Cavendish School of Computer Science, London, England, 2002 May 17, and at the University of Nottingham, School of Computer Science and Information Technology, Nottingham, England, 2002 May 20.

"A Finite Automata Model for Zulu Syllable Structure", joint lecture with Dr. Nhlanhla Thwala, Computer Science Seminar series, University of the Witwatersrand, 1999 November 12.

"Software: Engineering or Pre-Engineering? or How (not) to Blow Up the Ariane 5 and DM 1200 in Forty Seconds", Faculty of Science Lecture, University of the Witwatersrand, Johannesburg, South Africa, 1999 May 12.

"Algorithms to generate verification conditions: Some design issues and alternatives", Third Southern African Workshop on Theorem Proving and Model Checking (TPMC), University of the Witwatersrand, Johannesburg, South Africa, 1999 January 29.

"On the Inadequacy of Ordinary Preconditions for the Practical Design and Verification of Programs", International Symposium on Software Reliability Engineering (ISSRE '98), Paderborn, Germany, 1998 November 5.

"On the inadequacy of ordinary preconditions for the practical design and verification of programs", Colloquium lecture, Computer Science Department, University of Cape Town, 1998 July 9.

"A degree in software engineering — a start of a new discipline", lecture in the Computer Science Seminar series, University of the Witwatersrand, 1997 August 1.

"CS Education and an Engineering Approach to Software Development", lecture at McMaster University, Hamilton, Canada, 1997 July 24.

"Correct by Design: Mathematically Rigorous Program Development — an Example", lecture at McMaster University, Hamilton, Canada, 1997 July 24.

"Engineering vs. Pre-engineering or How software developers blew up the Ariane 5 and DM 1200 million in 40 seconds", lecture at McMaster University, Hamilton, Canada, 1997 July 22.

"CS Education and an Engineering Approach to Software Development", 27th Southern African Computer Lecturers' Association Conference (SACLA 97), 1997 June 22-24.

"No strict preconditions, no Ariane 5 or how software developers blew up DM 1200 million in 40 seconds", lecture in the Computer Science Seminar series, University of the Witwatersrand, 1997 March 7.

"Comparison of electrical 'engineering' of Heaviside's times and software 'engineering' of our times", lecture at the seminar "History of Software Engineering", Schloß Dagstuhl, Germany, 1996 August 26-30.

"Correct by Design: Mathematically Rigorous Program Development — an Example", lecture in the Computer Science Seminar series, University of the Witwatersrand, 1996 March 8.

“Mathematisch rigorose Programmkonstruktion: einige praktische Erfahrungen”, Vortrag, Workshop, DGLR Fachausschuß Software Engineering (T 6.6), Daimler-Benz AG, Stuttgart-Möhringen, 1995 November 29.

“Mathematisch rigorose Softwareentwicklung”, Kolloquiumsvortrag, IBM Heidelberg, 1995 Oktober 27.

“Zuverlässigkeit durch mathematisch rigorose Softwarekonstruktion”, Vortrag, Informatik-Kolloquium, Universität Erlangen-Nürnberg, 1995 Juli 17.

“Correct by design: simple, practical, mathematically rigorous program development — an example”, lecture at the seminar “High Integrity Programmable Electronic Systems”, Schloß Dagstuhl, Germany, 1995 February 27 - March 3.

“Mathematisch rigorose Programmkonstruktion: ein Beispiel”, Vortrag beim 1. ENCRESS Workshop, Institut für Sicherheitstechnologie (ISTec) GmbH, München, 1995 Januar 24.

“Qualität durch ingenieurmäßige Softwareentwicklung”, Vortrag bei der Euroforum Konferenz “Qualität in der Software-Entwicklung”, München, 1994 Dezember 8-9.

“Designing Provably Correct Software in Practice”, tutorial at the conference *Software Engineering — ESEC ’93, 4th European Software Engineering Conference*, Garmisch-Partenkirchen, Germany, 1993 September.

“Programmkorrektheitsbeweise: Theorie und Praxis, Konstruktion und Verifikation”, Vortrag in der VDI-GIS-ATZ Tagung “Zuverlässigkeit und Qualität von Software — Zufall oder bestimmbar?”, Hamburg, 1993 März 25-26.

“Erstellung fehlerfreier Software”, GI Regionalgruppe Rhein-Main, Darmstadt-Wixhausen, 1992 Januar 21.

“Neue Ideen und Konzepte für die Erstellung nachprüfbar fehlerfreier Software”, Vortrag im Symposium “Entwicklung und Prüfung sicherheitsbezogener Systeme”, veranstaltet von RWTÜV und TÜV Rheinland, Köln, 1990 Oktober 16.

“Some Personal Views of Software Design”, Special Lecture at the International Summer School on Programming and Mathematical Method, Marktoberdorf, Germany, 1990 July 25 - August 4.

“Neue Ideen und Konzepte für eine fehlerfreie Software-Erstellung — Was Software-Unternehmer und -Entwickler beachten sollten”, 3. Berliner Software-Unternehmer-Gespräch, 1990 Februar 9.

“Entwurf prüfbarer und korrekter Software”, Technische Hochschule Darmstadt, Praktische Informatik, 1990 Januar 25.

“Methoden zur Sicherung der Korrektheit von Computerprogrammen”, Kolloquium der Fakultät für Elektrotechnik, Universität Karlsruhe, 1988 November 24.

“Methoden zur Sicherung der Korrektheit von Computerprogrammen”, Jahrestreffen der Freunde des Fachbereichs Angewandte Informatik und Mathematik der Fachhochschule Fulda e.V., 1988 Mai 28.

“A General Overview of the Software Situation in the Federal Republic of Germany”, Institute of Software of the Academia Sinica (Chinese Academy of Sciences), 1987 July 27, Beijing.

“Software + Wartung = Widerspruch oder beliebte, bequeme Mythen entblößt”, Spektrum 86 Kongreß für Software-Maintainability, 1986 November 5-7, DECollege, München.

“The User Interface: Window to an Informational World”, User Interfaces International Conference, 1986 October 20-21, Gottlieb Duttweiler Institut, Zürich.

“Summary and Perspectives: Windows Revisited”, User Interfaces International Conference, 1986 October 20-21, Gottlieb Duttweiler Institut, Zürich.

“Software-Entwicklung: von ‘Stricken und Flicken’ zu einer Ingenieurwissenschaft”, an der Fachhochschule Fulda gehaltener Vortrag, 1985 November 5.

“Ein Überblick über informationstechnologische Entwicklungen bis 1995”, 6. Online Frühjahrstagung, Neu Isenburg, Bundesrepublik Deutschland, 1984 Mai 10.

“Overview of Information Technology Developments to 1990”, 7th International Online Information Meeting, London, England, 1983 December 7.

“Software Development: Science or Patchwork?”, in the series “van specificatie tot implementatie”, Informaticacolloquium of the Centrum voor Wiskunde en Informatica (CWI), Amsterdam, The Netherlands, 1983 November 18.

“Softwareentwicklung gestern, heute und morgen — Wissenschaft oder Flickwerk?”, in der Vortragsreihe über Zusammenhänge der Mathematik mit anderen Wissenschaften, Mathematisierungskolloquium der Gesamthochschule Kassel, Bundesrepublik Deutschland, 1983 Mai 13.

Dr.-Ing. Robert Laurence Baber

Research interests

Premise: Software development is by nature an engineering discipline. A theoretical and mathematical foundation for software development, comparable to the scientific foundations of the traditional engineering fields, exists but has not yet been embraced to any significant extent by software development practitioners. This deficit is the main factor preventing software from assuming as dominant a role in, for example, safety critical applications and systems as artifacts of the traditional engineering disciplines.

Overall goal: Contribute to the metamorphosis of software development practice to a truly engineering discipline.

Background: The many reasons cited for not employing in practice the existing mathematical basis for software development and verification ("formal methods", correctness proof techniques, etc.) converge in effect in the perception that the cost of acquiring and employing the ability to use them in practice is too high and uneconomical. The mathematics is often said to be too difficult for software developers to learn; their use by those capable of understanding them requires too much time and effort. "Tools" are expected by some to be the solution to this problem, but there is no reliable and significant promise of their emergence on the commercial and industrial scene in the foreseeable future.

Strategy: Identify specific inhibiting factors and develop solutions or remedies. Extend on the work of Baber, Parnas and others in developing an approach for specifying, designing and analytically verifying software that

- is aimed towards practitioners and their needs, that
- reflects an engineering mentality and that
- is systematic and repeatable, e.g. by employing standardized formats and approaches for formulating and documenting requirements, interfaces, etc., using various representational forms such as text, tables, diagrams, formulae, etc.

Role models: The transformation of other technical fields to engineering disciplines, e.g. electrical, mechanical, civil, chemical engineering, etc.

My research: My research to date has concentrated on various activities of the types outlined above. I am interested in the problem of identifying components of the standard body of knowledge which all future software engineers will be required to know and developing appropriate teaching materials and methods for undergraduate and graduate software engineering programs as well as courses and seminars for software developers already in commercial practice. I have recast selected parts of the known theoretical basis for proving computer programs correct and have filled gaps in that theory that limited its practical application too severely. I have formulated a complete set of "proof rules" (lemmata) that enables software development practitioners to prove the total correctness of their programs (i.e. including proving the absence of run time errors) and that enables them to design correct software.

Future work: In the future I would like to work on

- "repackaging" selected theoretical material on ensuring the correctness of programs with a view to making it easier to apply in practice (e.g. by simplifying and consolidating the mate-

rial and its presentation, by refining, standardizing and developing new techniques for applying it, etc.),

- transferring the knowledge and abilities useful for practical application to software development practitioners, e.g. through seminars, reports, self-study materials, on-line courses, etc.,
- assisting future teachers of such material, e.g. by developing and instructing teach-the-teachers seminars and courses, by supplying teachers with instructional materials, advice, etc., and
- improving and extending already known techniques for designing programs by deriving large parts of them from their specifications and correctness proof considerations, much as engineers in the traditional disciplines derive many aspects of their artifacts from mathematical models based on the relevant underlying scientific and mathematical theory (e.g. F=ma, Maxwell's equations, Kirchoff's voltage and current laws, material and energy conservation principles, the laws of thermodynamics, etc.).

Computerized support is often proposed as a necessary prerequisite for employing in practice mathematically based methods for ensuring the correctness of computer software. While I disagree that it is an essential prerequisite, there can be no doubt that computerized support oriented towards the needs of the practitioner (in contrast to those of the researcher) and of "industrial strength" could be of considerable practical value. I would, therefore, also be interested in assisting in the design and implementation of such systems, e.g. by helping to investigate the internal design issues, technical requirements, desirable features and possibilities of such computerized support. The goal of such research could, for example, be to determine specific and detailed technical guidelines for designing and implementing commercial grade computerized support for program verification. Commercial software development organizations could then develop corresponding products with considerably less risk.