Ramesh Bharadwaj

Naval Research Laboratory, Code 5546, 4555 Overlook Ave. SW, Washington DC 20375
http://www.seas.gwu.edu/~ramesh, ramesh@itd.nrl.navy.mil, (202) 767-7210

Research Summary:

My research focuses on both the Foundational *and* Practical aspects of Software Engineering. Foundational areas include languages and methods for specification & verification of systems, model checking, combining decision procedures, and concurrency theory. On the practical side, I *build* tools to support the specification and analysis of embedded real-time systems.

I am interested in all facets of Software Engineering: requirements specification, methods and notations for design such as UML, automatic testing and test case generation, implementation (esp. of distributed systems using Java/CORBA), and software project management. I have hands-on experience in systems development, having worked in industry as a hardware engineer, software engineer, senior systems programmer, and software development manager.

I have provided leadership in the development and teaching of graduate and undergraduate courses in computer networking, software engineering, concurrent programming, protocol verification, and programming in Java^{TM}.

Education:

Ph. D. in Software Engineering, McMaster University, Canada, Mar 1996

Thesis: A Method and Supporting Tools for Systems Verification *This thesis is an attempt to bring the craft of software construction closer to an engineering discipline. We transform theoretical ideas in program verification, concurrency theory, and automatic theorem proving, into a concrete method for the specification and analysis of software artifacts. We also describe tools being developed to support our method.*

Master of Electronics Engineering, PII, The Netherlands, May 1982

Project: A CAD System for Digital Filter Banks *This thesis describes a computer aided design tool and special purpose hardware, developed to automate the implementation of minimum-norm recursive digital filter banks. The tool supports parameter selection, simulation, normalisation, discretisation, and directly generates microcode for the hardware.*

Bachelor of Electrical Engineering— Mysore University, India, May 1980

Project: Design of an SSB/SC HF Transceiver

Current Employment:

Computer Engineer 1997 to current
Naval Research Laboratory, Washington, DC— Scientist in the Software Engineering Section conducting research in methods and tools for the specification, design, and analysis of software-intensive systems for embedded applications.

Adjunct Faculty 1997 to current
The George Washington University, Washington DC— In my software engineering and advanced object oriented programming courses I teach how a formal approach to software development improves quality and reduces costs in industrial software development.

Previous Employment:

Senior Software Engineer 1995 to 1997
Kaman Sciences Corporation (at the Naval Research Laboratory)— The SCR method for the specification and analysis of hybrid systems. Tools to support software development for embedded real-time systems.

Research Assistant 1991 to 1995
Communications Research Laboratory, McMaster University— Tools, languages, and methods for Software Engineering.

Member Technical Staff 1993 to 1994
AT&T Bell Laboratories, Murray Hill, NJ— Development of formal methods and tools for the verification of telecommunications software and hardware.

Senior Systems Engineer 1993
AT&T Network Systems, Richmond, VA— Design and development of a visual language and application development tools for an information services platform.

Manager, Systems and Networking 1990 to 1992
The Research Libraries Group, Inc., Mountain View, CA— Technical and manpower management of development projects for the design, standardisation, reference implementation, and conformance testing of ISO protocols for information retrieval.

Senior Systems Programmer 1988 to 1990
RLG, Stanford University, Stanford, CA— Design and implementation of end-to-end protocol stacks for wide area networks, including the OSI transport protocol TP4, network tunnels for internet protocols over X.25 packet switching networks, and the design and implementation of application protocols for information retrieval.

Senior Software Engineer 1987 to 1988
Networking Division, SYSTIME Computers Ltd. — Senior project member responsible for the development of internetworking solutions (over SNA, OSI, and Decnet networks) for major US clients.

Staff Scientist 1982 to 1987
Real-Time Systems Group, NCSDCT, Tata Institute, Bombay, India— Team member responsible for the design and implementation of multi-access protocols for broadcast packet switching networks, design of corporate data networks for several major corporations, tools and methods for protocol verification and standardisation, and the development of network components (XXX-compliant PAD and low-end switch) for a packet switching network.

Researcher/Summer Intern 1981 to 1982
Networking and Distributed Systems Group, Philips Research Laboratories— design and development of hardware for an Ethernet controller; development of a computer aided design tool for synthesis of digital filter banks for voice recognition (delivered to Philips Research, Hamburg).

Assistant Engineer(hardware) 1980 to 1981
Research and Development Division, Hindustan Computers Ltd., New Delhi— board-level design of a hard disk controller for the HCL-8S scientific computer.

Refereed Journal Publications:

Tools for Tabular Specification and Analysis of Requirements. Formal Methods in System Design (submitted).

Applying the SCR requirements method to the light control case study. Journal of Universal Computer Science, July 2000.

Model checking complete requirements specifications using abstraction. Journal of Automated Software Engineering, Jan 1999.

Using abstraction and model checking to detect safety violations in requirements specifications. IEEE Transactions on Software Engineering, Nov 1998.

Refereed Conference Papers:

Developing High Assurance Avionics Systems with the SCR Requirements Method. in *Proc.* 19^{th} *Digital Avionics Systems Conference*, Philadelphia, PA, October 2000.

Salsa: Combining Constraint Solvers with BDDs for Automatic Invariant Checking. in *Proc. Tools and Algorithms for the Construction and Analysis of Systems (TACAS'2000)*, Berlin, Germany, March 2000.

Hardware/Software Co-Design and Co-Validation Using the SCR Method. In *Proc. IEEE Int'l High Level Design Validation and Test Workshop (HLDVT'99)*, San Diego, CA, November 1999.

A toolset for specifying and analyzing software requirements. In *Proc. Computer-Aided Verification, 10th Annual Conf. (CAV'98)*, Vancouver, BC, Canada, June 1998.

Applying the SCR requirements method to a simple autopilot. In *Proc. Fourth NASA Langley Formal Methods Workshop*, Hampton VA, USA, Sept 1997.

Verifying SCR requirements specifications using state exploration. In *Proc. First ACM SIGPLAN Workshop on Automatic Analysis of Software*, Paris France, January 1997.

Applying the SCR requirements specification method to practical systems: A case study. In *Proc.* 21^{st} *Sofware Engineering Workshop*, NASA GSFC, Greenbelt, MD, Dec 1996.

Formalizing inductive proofs of network algorithms. In *Proc.* 1^{st} *Asian Computing Science Conference*, Pathumtani, Thailand, December 1995.

Electronic document transmission over the Internet. In *Proc.* 11^{th} *Annual Conference of the American Society for Information Science (ASIS)*, Washington DC, October 1991.

CSP as a formal description technique for OSI protocols. In *Proc. IBERICOM-87*, Lisbon, Portugal, February 1987.

Invited Talks:

An Approach for Consistency Checking SCR Requirements Specifications, *ONR C2 and Combat Systems Gathering.* Naval War College, Newport, Rhode Island, April 24-28, 2000 (poster session).

Analysis of Agent-Based Systems Using Decision Procedures, *1^{st} Goddard Workshop on Formal Approaches to Agent-Based Systems.* NASA Goddard Space Flight Center, Greenbelt, Maryland, April 5-7, 2000 (poster session).

Rigorous Requirements for Safety-Critical Systems: Fundamentals and Applications of the SCR Method, *European Joint Conference on Theory and Practice of Software 2000.* Berlin, Germany, March 26, 2000 (full day tutorial).

Model Checking and Beyond: The Verification of Software Requirements Specifications, *IFIP WG 2.9 Workshop on Software Requirements Engineering.* Flims, Switzerland, February 5-9, 2000.

An SCR profile for the UML, at the Analysis and Design Task Force meeting of *The OMG Technical Committee* in Mesa, Arizona January 2000.

Analysis of SCR Specifications Using Decision Procedures, *Dagstuhl Seminar on Requirements Capture, Documentation, and Validation.* Schloss Dagstuhl, Germany June 1999.

Automatic Code Generation from Requirements Specifications, *Automatic Code Generation and the Software Development Process.* Boeing Corporation, Washington DC June 1999.

Salsa: Combining Decision Procedures for Fully-Automatic Verification, *Computer Science Department, University of Texas at Dallas*, April 1999.

Requirements Based Testing – A Research Agenda, *8^{th} Annual Automated Software Test and Evaluation Conference.* Washington, DC March 1998.

Verification of SCR specifications, *5^{th} Int'l Software Cost Reduction Workshop.* Bell-Northern Research Ltd., Ottawa, Ontario Canada, Feb 1996.

Terminal servers for private X.25 networks, *International Telecommunications Technology Forum*, Ft.Lauderdale, Florida, April 1991.

COMNEX – An experimental satellite packet broadcast network, *Workshop on Satellite Communication*, Space Applications Centre, Ahmedabad, India, August 1986.

Technical Reports:

Salsa: Combining decision procedures for fully automatic verification. In preparation.

Model Checking Complete Requirements Specifications Using Abstraction, Memorandum Report NRL/MR/5540–97-7999, Naval Research Laboratory, Washington DC, November 1997.

Tools to Support a Formal Verification Method for Systems with Concurrency and Nondeterminism, CRL Report No. 327, McMaster University, Hamilton Ontario, Canada, April 1996.

Direct Model Checking of Temporal Properties, CRL Report No. 317, McMaster University, Hamilton Ontario, Canada, January 1996.

Snap: A Proof Development System for First-Order Logic, Technical Report CRL 302, McMaster University, Hamilton ON, Canada, May 1995.

Towards a Checker/Annotator for Proofs of Distributed Programs, Technical Memorandum, AT&T Bell Laboratories, Murray Hill NJ, November 1994.

TOP/Snap: A Transition Oriented Prover for PROMELA, Technical Memorandum, AT&T Bell Laboratories, Murray Hill NJ, August 1994.

The Design of a Communications Server, Technical Report, The Research Libraries Group, Inc., Mountain View CA, April 1991.

EtherTerm: Tunneling TCP/IP over X.25 Networks, Technical Note, The Research Libraries Group, Inc., Mountain View CA, Oct 1990.

EtherChannel: An Ethernet Channel Interface for the IBM3090, Technical Report, The Research Libraries Group, Inc., Mountain View CA, March 1989.

An Adaptive Contention Technique for Random Access to Broadcast Channels, TR-43, NCSDCT, TIFR, April 1986.

Computer Aided Design of Digital Filter Banks, Philips Technical Report, Natuurkundig Laboratorium, Eindhoven, June 1982.

Refereeing:

- *Computer Aided Verification*, 1994.
- *Protocol Specification, Testing and Verification*, 1994.
- *FORTE*, 1994.
- *ICSE*, 1996, 2000.
- *COMPASS*, 1996, 1997.
- *DCCA*, 1998.
- *TACAS*, 2000.
- *IEEE Computer.*
- *IEEE Transactions on VLSI Systems.*
- *IEEE Transactions on Software Engineering.*
- *Formal Methods in System Design.*
- *Requirements Engineering Journal.*

Teaching Highlights:

`Advanced Object-Oriented Programming:`
The George Washington University, the emphasis of this course is on a fundamental tenet of practical programming – *reuse*. We demonstrate reuse in the context of Design Patterns and the C^{++} Standard Template Library (STL).

`Software Engineering:`
The George Washington University, a full semester graduate level course which teaches how formal specification and verification tools can play a useful role in industrial software development, especially for debugging system specifications and design *before they are built*.

`Web Authoring using Java:`
The George Washington University, a full semester graduate level course which emphasises software engineering and object oriented design issues in the creation of interactive content for the web using the Java programming language.

`Protocol Verification:`
Swedish Royal Institute of Technology, a full credit course which presented a method and tool (PROMELA/SPIN) for the automatic validation of protocols, concluded by an exercise in validating a "real life" protocol — the Internet standard protocol **PPP**.

`System Software:`
National Centre for Software Technology, Bombay, a full semester course which explained the rudiments of language processing including compiler construction, and operating system principles, culminating in the design and implementation of a language processor and a process scheduler.

Other Courses:

Full Semester Courses
- Computer System Organisation
- Programming Techniques
- Computer Organisation and System Software
- Software Engineering (Teaching Assistant)
- Introductory Programming (Teaching Assistant)

Short-term Courses:
- Computer Networking
- Software Reliability
- Distributed Computing
- Concurrent Programming

Thesis Guidance: Supervised a Master's Thesis project for the Department of Computer Science, Indian Institute of Technology, Bombay on *The Design and Implementation of a 68000-based packet switch*.

Examinership: Served as external examiner for graduate and undergraduate thesis defences, at the Department of Computer Science, Indian Institute of Technology, Bombay.

Development Highlights:

`Salsa`

I architected, designed, and managed the implementation of *Salsa*, a tool for the automated analysis of specifications in SAL (the **S**CR **A**bstract **L**anguage). Salsa's primary verification engine is an *invariant checker*, which may be used to check a SAL specification for unwanted nondeterminism and missing cases, for the verification of properties formulated by users, and for automatic test case generation. We have used Salsa to find real bugs in Navy systems.

`The SCR* Toolset`

Added a verification capability to the SCR* toolset, which allows users to verify safety properties of SCR specifications using the model checker Spin. State explosion is minimised by verifying abstractions of the original requirements specification, which are derived by using two correctness-preserving transformations.

`Ariel`

Architected, managed and led a project team which developed a product for transmitting electronic documents over the Internet. Ariel is a stand alone system which scans, manipulates, transmits and prints documents, in addition to interoperating with CCITT T.30 compliant systems. Ariel is now in use at more than 10,000 institutions around the world.

`Private Packet Network:`

Senior member of a project team which architected, designed, developed, and deployed a nationwide private packet switching network, connecting more than 1200 diverse terminals and host computers, which uses TCP/IP, OSI and X.25 communication protocols.

`ERNET:`

Designed and proposed the development of a packet switching network for educational and research institutions in India, which culminated in a UNDP-funded multi-institution project ERNET, which currently provides Internet connectivity to universities and research institutions in India.

`PakNet:`

Architected and developed a modular data network which used diverse media (PSTN, satellite channels, VHF/UHF radio channels and microwave links) to provide X.25 compliant interfaces to user equipment and host computers.

`COMNEX:`

Member of a project team which designed and implemented an experimental packet broadcast network for random access communication over SCPC satellite channels.

Other Systems:

- Architected and implemented a table holder module.
- Application development tools for an information services platform.
- Implementation of algorithms for skeletonising digital images.
- Design of neural networks for supervised and unsupervised learning.
- An OSI based record transfer and information retrieval protocol.
- Implementation of a protocol description language and simulator.
- Implementation of a compiler for CSP (Hoare 1978).
- Hardware and MAC layer software for an Ethernet controller.

Networking Software:	Development of cluster terminal controllers, front-end software and network management tools for an X.25/OSI based private packet switching network. Software and hardware development for X.25 packet switching nodes and CCITT XXX compliant PADs.
Software Tools:	Cross development environment for packet switching software, including a Modula-2 cross compiler, cross & native linkers and a post-mortem debugger. Development system for a single-board computer – cross macro assembler, debugging monitor, real-time kernel, downline loader and EPROM programmer.
Protocol Design:	OSI-compliant protocols for information retrieval. An Adaptive Contention Protocol for efficient shared access to satellite packet broadcast channels.
Professional Societies:	Member of the ACM and the IEEE Computer Society.
Honours and Awards:	On-the-spot-award for exceptional job performance, February 2000. Alan Berman Research Publications Award, 1998. Philips Scholarship, 1981.
References:	Prof. David L. Parnas, *(Ph.D. thesis supervisor)* NSERC/Bell Industrial Research Chair in Software Engineering Director of the Software Engineering Programme Department of Computing and Software Faculty of Engineering McMaster University Hamilton ON, Canada L8S4K1 Email: `parnas@mcmaster.ca` Prof. Jeffery I. Zucker, Department of Computing and Software McMaster University Hamilton ON, Canada l8S4K1 Email: `zucker@mcmaster.ca` Prof. Rance Cleaveland II, Department of Computer Science State University of New York at Stony Brook Stony Brook, NY 11794 Email: `rance@cs.sunysb.edu` Dr. Gerard J. Holzmann, Computing Sciences Research Center Lucent Technologies Bell Laboratories 600 Mountain Avenue Murray Hill, NJ 07974 Email: `gerard@bell-labs.com`