

David James Benn
73 Second Avenue, Klemzig, SA, 5087
08 8266 6508 [home]
0407 261163 [mobile]
dbenn@computer.org
<http://www.users.on.net/~dbenn/>

Career Summary

A Software Engineer, with more than a decade of experience in a variety of domains, from embedded systems to desktop software.

Possesses significant experience and knowledge of all aspects of the software engineering process.

Acted in lead technical roles on projects, mentoring less experienced engineers, driving product and capability roadmaps, liaising with customers.

A focus upon quality, involved in identifying best practices, technologies, tools, procedures and standards.

Key Strengths

- Extensive skills in design, coding, unit testing, reviews, system testing, requirements gathering and analysis, and configuration management.
- Experienced in a range of domains and software development methodologies.
- Experienced in object-oriented design, languages, and design patterns.
- Team oriented, encouraging the use of common procedures and practices.
- Test focussed, emphasising the importance of regression and repeatability.
- A strong commitment to life-long learning, enjoying teaching and mentoring.
- Innovative in problem solving.
- Excellent written and verbal communication skills.

Domains

- Embedded software (smartcards, mobile phones)
- Development, test, and verification tools
- GUI redesign and maintenance
- Relational databases and object- relational mapping
- Hardware simulation

Programming Languages and Tools

- Java 5, C, C++, Gnu compiler tool-chains
- Perl, Bourne shell, some experience with Groovy, Python, Ruby
- SQL, JDBC, Hibernate
- XML standards and technologies including XSLT, DOM, DTD
- Eclipse (for Java and C++)
- emacs, vi
- JUnit, JCoverage
- CVS, Subversion, RCS, Clearcase

- Ant, make
- ANTLR compiler construction language
- Non-commercial experience with ECMAScript (JavaScript), AJAX, PHP
- Brief commercial exposure to .NET (C#, Visual Studio 2005, Nant, NUnit, NCover) with other non-commercial .NET experience
- Borland Together and Rational Rose for object-oriented analysis and design
- Test coverage, memory leak detection, lint, profiling tools for C, C++, Java
- Processor- specific simulators (eg M*CORE)
- VNC, VMWare
- FrameMaker, Microsoft Office, Open Office

Professional Experience

Australian Semiconductor Technology Company, Adelaide Jun 2007 – Present time

Software Developer

Development and test of simulation components for mobile phone Systems on a Chip (SoC) in C++.

Achievements

- Development of multiple SoC component models and C driver code.
- Bare-metal ARM simulator development and testing.

Internode, Adelaide

May 2006 – Jun 2007

Analyst Programmer

Design, code, and test of internal Java and Perl software systems, primarily payment processing related.

Achievements

- Development, unit, and integration testing with Java 5, Eclipse, JUnit, Maven, Cobertura, Hibernate, Tapestry, Spring, Perl.
- Remained close to estimates.

Worldsmart Technology, Adelaide

Apr 2006

Senior Software Engineer

Custom WinCE device driver and C# (.NET 2.0) development for Adelaide hotel industry.

Achievements

- Introduced a test-driven approach and tool flow (Nant, NUnit, NCover, TestDriven.NET, suggested CruiseControl.NET).
- Played a leading role in finding and fixing a non-deterministic device driver bug that had eluded others.

**Freescale Australia SoC Technology Centre, Adelaide
2006**

Jul 2004 – Mar

Senior Staff Software Engineer

Motorola's Semiconductor Products Sector was spun off as a separate company in Dec 2004, initially as a subsidiary of Motorola in April 2004, with the name of Freescale. Freescale designs and manufactures a variety of integrated circuits (ICs) and Systems on a Chip such as 68000 and PowerPC derivatives.

Achievements

- Worked in a US-Australian team to develop an object- relational mapping layer to replace an existing, costly, and undocumented layer, as part of an IC design verification tool. Used Java 2, JDBC, MySQL, CVS, Eclipse, Ant, Groovy. Applied Gang of Four design patterns, and data access patterns.
- Worked with a colleague to develop a regression test suite as part of this work, and with the

rest of the team to improve collaborative development and software quality practices (primarily testing and change control). Used JUnit, JCoverage.

- Spent two weeks visiting and working with the rest of the team in Austin, Texas.
- Provided support and training to Adelaide project team members using this IC design verification tool.
- Improved the software development capability of Freescale Australia by helping to organise and presenting at technical discussions, writing application notes, and encouraging cross-project knowledge sharing.

Motorola Australia Software Centre, Adelaide
2004

Apr 1997 – Jun

Senior Staff Engineer

Achievements

- Developed part of a formal verification tool for embedded system model-based designs on a 10 person team. C++ and STL were used along with ANTLR for compiler development. Used the Borland Together tool for design and limited round-trip engineering in this and other projects at Motorola.
- Redesigned the GUI for this formal verification product so that it could be used with three variants of the tool. Used Java and the Swing library for GUI development. The efficiency and maintainability of the GUI code base were also improved.
- Worked with 3 others to develop an industrial strength SAT solver for use in design verification (primarily hardware) by studying the literature. C++ and STL were used. Understanding the computational complexity of proposed implementations was particularly important in order to maximise speed and minimise memory usage. Used Perl and Python for testing, debugging and other utilities. Used Gang of Four design patterns here and in other 2 projects above.
- Worked on a 25 person smartcard operating system project as a technical lead and developer. An embedded systems project with extreme resource limitations and security requirements. C was the main language, with some M*CORE (32-bit RISC) assembly used for time-critical code.
- With a team of 6, prototyped a JavaCard smartcard development and testing environment, then as technical lead of a smaller group created a JavaCard virtual machine for a product based upon this prototype. These were demonstrated at trade shows and tested by potential customers. In the latter stages, took on the role of Software Architect to coordinate three technical leads and plan for the product's future with an eventual team of 25 developers.
- Worked with 2 other developers to prototype the modification of a Motorola mobile phone operating system in order to interface with a barcode reader. C was the development language.
- Worked alone to create a smartcard reader protocol stack in C (Visual Studio 6).
- Took on a project lead role for a team of 7 software engineers to develop a WAP-based system. Project plan development, customer liaison, reporting, and scheduling were the primary responsibilities.

University of South Australia

2001 – 2005

Sessional tutor and guest lecturer

Wanted to maintain contact with the University to share industry experience with students, and maintain contact with University research and teaching trends.

Achievements

- Delivered reverse engineering lecture, Information Systems Maintenance and Re-engineering subject (2004, 2005, 2006).
- Tutored for a first year Programming in Java course (2005).

- Lectured and tutored in Intelligent Systems Technology (Lisp programming, 2001, 2003, 2004).

Vision Internet

Dec 1995 – Apr 1997

Unix System Administrator and Programmer

- Vision Internet's system administrator:
 - Responsible for the day-to-day running of the BSDi and FreeBSD Unix Systems of Vision Internet Services;
 - Ensured the integrity, availability, and confidentiality of the data in my care. Vision had in excess of 1000 users by around mid-1996. Other duties included installation of a variety of Unix programs as needed, applying security patches, providing user support, liaising with Federal police and AUSCERT during a security breach.
 - Vision Internet's programmer, working primarily with C, Perl, and shell scripts. Wrote custom CGIs for use with web pages, and miscellaneous system utilities as required.
 - Primary software development was an online accounting system which worked with the raw Unix and Cisco terminal server accounting logs, handled monthly invoicing, daily time debiting, session termination, and general time-usage administration and notification. This was an ongoing project since Vision's account types were fluid, and the company's state-wide presence continued to grow. This software was also sold in 1996 to another Tasmanian Internet provider, and custom updates were provided as necessary.

University of Tasmania

Feb 1993 – Nov 1995

Level A Academic in Applied Computing

- Taught a variety of undergraduate computing units. Developed course structure and teaching material for several of these units.
- Mostly small classroom and computer laboratory situations, but also developed and delivered lectures for computer security, programming paradigms units.
- Assessed work carried out by students during the semester and in formal examinations. Designed numerous assignments and in a few cases, final examinations. Second assessor for other units' exams.
- Participated in departmental research groups including Multimedia and Expert Systems, although teaching was the main focus.
- Contract was not renewed due to funding cuts, despite a recommendation by the head of department for another 3 year contract.
- Taught the following computing subjects during this period:
 - Programming and Problem Solving with Pascal
 - Data Structures and Algorithms
 - Computer Security
 - Introduction to Operating Systems, Principles of Operating Systems
 - Expert Systems
 - Programming Paradigms
 - Computer Concepts (transistors to processors to assembly language to operating systems)
 - Computing Systems (word processing, spreadsheets, databases)
 - Computer Networks and Digital Communications

University of Tasmania

Feb 1992 – Dec 1992

Part-time tutor in Applied Computing

- Enjoyed the minor teaching component of CSO position below, so chose to take up a tutoring position.

University of Tasmania**Jul 1991 – Feb 1992*****Computer Systems Officer***

- Computer Systems Officer (CSO) for the Technical Services section of the Department of Applied Computing with wide-ranging duties: Macintosh and PC software support, hardware (including network) installation and maintenance, Novell Network administration, Unix
- Primary software development was an online accounting system which worked with the raw Unix and Cisco terminal server accounting logs, handled monthly invoicing, daily time debiting, session termination, and general time-usage administration and notification. This was an ongoing project since Vision's account types were fluid, and the company's state-wide presence continued to grow. This software was also sold in 1996 to another Tasmanian Internet provider, and custom updates were provided as necessary.

University of Tasmania**Feb 1993 – Nov 1995*****Level A Academic in Applied Computing***

- Taught a variety of undergraduate computing units. Developed course structure and teaching material for several of these units.
- Mostly small classroom and computer laboratory situations, but also developed and delivered lectures for computer security, programming paradigms units.
- Assessed work carried out by students during the semester and in formal examinations. Designed numerous assignments and in a few cases, final examinations. Second assessor for other units' exams.
- Participated in departmental research groups including Multimedia and Expert Systems, although teaching was the main focus.
- Contract was not renewed due to funding cuts, despite a recommendation by the head of department for another 3 year contract.
- Taught the following computing subjects during this period:
 - Programming and Problem Solving with Pascal
 - Data Structures and Algorithms
 - Computer Security
 - Introduction to Operating Systems, Principles of Operating Systems
 - Expert Systems
 - Programming Paradigms
 - Computer Concepts (transistors to processors to assembly language to operating systems)
 - Computing Systems (word processing, spreadsheets, databases)
 - Computer Networks and Digital Communications

University of Tasmania**Feb 1992 – Dec 1992*****Part-time tutor in Applied Computing***

- Enjoyed the minor teaching component of CSO position below, so chose to take up a tutoring position.

University of Tasmania**Jul 1991 – Feb 1992*****Computer Systems Officer***

- Computer Systems Officer (CSO) for the Technical Services section of the Department of Applied Computing with wide-ranging duties: Macintosh and PC software support, hardware (including network) installation and maintenance, Novell Network administration, Unix and VAX/VMS system administration, Unix, Mac, and PC programming (C and Pascal), acting as a resource person on technical matters for students and staff, running short courses on various aspects of Unix.

- University of Tasmania** **Feb 1991 – Jun 1991**
Trainee Computer Systems Officer
- Worked with a variety of computing systems (Mac, PC, Unix, VAX/VMS), wrote programs for teaching purposes, and laid the groundwork for becoming a jack-of-all-trades CSO.
- Registered Nurse, Launceston General Hospital** **Feb 1987 – Jan 1990**
- Intensive/Coronary Care, Operating Room, and Recovery Room.
- Registered Nurse, Kimba District Hospital** **Dec 1986 – Jan 1987**
- Geriatrics, Casualty, Outpatients.
- Various employers through Adelaide agencies** **Feb 1985 – Jun 1986**
- Geriatrics, Private Medical/Surgical, High Dependency, Intensive Care.
- Registered Nurse, Queen Elizabeth Hospital** **Feb 1984 – Feb 1985**
- Urology, Ear-Nose-Throat, Neurology, Neurosurgery, Ophthalmology, Rheumatology, Recovery.
- General Nurse Training, Queen Elizabeth Hosp** **Feb 81 – Dec 1983**
- Gained the experience and qualifications necessary to practice as a Registered General Nurse.

Qualifications

- Master of Science (Computer and Information Science)** **2001**
 University of South Australia.
- Bachelor of Applied Computing** **1992**
 University of Tasmania.
- General Nursing Certificate** **1984**
 Queen Elizabeth Hospital.

Professional Memberships and Affiliations

- Member of the Australian Computer Society (ACS).
- Affiliate member of IEEE Computer Society.
- Member of Association for Computing Machinery.
- South Australian PIC microcontroller User Group.
- SA chapter of the Australian Java User Group.
- Australian .NET User Group, SA branch.

Continuing Education

- Joint editor of Second Conceptual Structures Interoperability Workshop.
- Invited to be an AI 2006 and 2007 reviewer (accepted).
- Presented papers at the International Conference on Conceptual Structures 2001 at Stanford University (<http://www.users.on.net/~dbenn/Masters/>). Accepted invitations to be on the Program Committee for ICCS 2002 and 2003.
- Attendance at ACS, PIC, Java, and .NET group meetings.
- Subscriptions to Dr Dobbs, IEEE, ACM, and ACS publications.
- Training from 2003 until now:
- ACM Introduction to SQL;

- Introduction to Microsoft .NET development (2 days, Kaz Training);
- RTL hardware design and verification with Verilog (2 days, in-house);
- MySQL training (1 week certification level course, MySQL in US);
- Verilog assertion- based functional verification training (1 day, in-house);
- Functional verification of Verilog using OpenVera (3 days, in-house);
- DotNetNuke training (1 day, Kaz Training, Australian .NET User Group);
- OO design, design patterns, advanced C++ (in-house, 3 half days);
- TI MSP430 day seminar (July 2006).
- Other training from 1997 to 2002: Structured Analysis and Design, RTM (requirements traceability and management tool), Tcl/Tk scripting, Project management, Estimation techniques, Capability Maturity Model, External Rational seminars on Configuration Management, OOAD, RUP, Embedded Design Automation domain training, Object-Oriented Analysis and Design with UML, Requirements Analysis with Use Cases, Standard Template Library, Extreme Programming, Time management.

Personal Projects

- C library of PIC microcontroller driver functions for LCDs, keypads, Dallas Semiconductor 1-wire protocol, 7-segment display multiplexing, state machines, timers, switch input, and so on. The code has been written in such a way as to be easily ported to other microcontrollers such as AVR. Various applications of the library, e.g. dimming night light using pulse width modulation, temperature logger. See <http://sourceforge.net/projects/picplib>
- Free compilers and interpreters:
 - ACE Basic for the Amiga. Generates 68000 assembly (1991–1996). Favourably reviewed in several magazines. Had an active community.
 - LittleLisp for the Newton PDA (2000).

See <http://www.users.on.net/~dbenn/docs/projects.html> for others.

Interests

- Spending time with wife Karen, son Nicholas (7), and daughter Heather (4). Bicycle riding, playing golf.
- Amateur astronomy. Member of the Astronomical Society of South Australia.
- Reading science, philosophy, history, science fiction.

Referees

Available upon request.