

Curriculum Vita

Current Research

Aspect nesC (U of U, Eric Eide and Robert Kessler, '05 - '07) – Aspect-oriented programming (AOP) augmentation of nesC, a componentized version of C, for sensor network programming and an exploration of AOP in componentized embedded systems.

Academic Publications

“Resource Management Aspects for Sensor Network Software.” Sean Walton and Eric Eide. PLOS 07.

“Feature Types in Early Aspects.” Sean Walton and Eric Eide. OOPSLA 05, Early Aspects Workshop.

Development of Event-Managed Process Scheduling (BYU, CS Master's Thesis/Project 1990) – A proposal and prototype for computer-language modifications to include methods for starting, stopping, and terminating processes and threads.

“Process Migration and Resource Affinity on a Transputer Multiprocessor” (BYU, Research Project 1989) – A discussion of issues and proposal for tagging and migrating processes towards resources or each other to increase processing power.

Research Positions

Network OS (BYU, Aurel Cornell, 1989) – Reviewed and analyzed Transputer project to determine methods for increasing parallel processing capability.

Senior Consultant (Carolla Development, 1997-1998) – Researched software engineering processes for reliable, “defect-free” projects.

Senior Architect (Nationwide Financial, 1998-1999) – Designed and oversaw cradle-to-grave process for software project development; solved the UML-database link.

Senior Consultant (Nationwide Insurance, 2000) – Researched new technologies for integration into corporate culture; initiated pilot projects to discover feasibility.

Senior Consultant (American Electric Power, 2000-2003) – Researched new/emergent technologies, made recommendations for corporate adoption and policy, negotiated contracts with product vendors, initiated pilot projects, and led PDA adoption integration.

Teaching/Lecturing Positions

Teacher's Assistant (BYU 1984-1989) – Taught individuals & groups, graded exams & homework, contributed to exams, held help forums, built tools to help students.

Instructor (UNISYS, 1988) – Taught intermediate and advanced courses on UNIX System V minicomputers.

Professional Lecturer (Carolla Development, 1998) – Built/taught courses on requirements gathering, OOA/D, C++, and Java. Received certifications for professional lecturing and course construction.

ACM Lecturer (Carolla Development, 1998) – Presented the process of how HP LaserJets automatically determine a job printer language, based on patented process.

Adjunct Faculty (OSU CIS, 1998) – Taught CIS214 (data structures), prepared exams, projects, and homework assignments.

Academic Courses Taught

Computer Languages (BYU CS330) – A broad sweep of the different types of computer languages, including C, Pascal, Lisp, Basic, APL, and Modula.

Introduction to Computer Architecture (BYU CS380) – A general study of the basic architectures of current computers, focusing primarily on the Von Neumann architecture.

Compiler Theory (BYU CS431) – An in-depth analysis and design of basic parsers and compilers. Included building a native-code generating parser-translator with arrays, procedures, control structures, and records.

Introduction to Operating Systems (BYU CS340) – An introduction to operating system shells and basic operating system functions.

Operating System Design (BYU CS440) – An in-depth coverage of operating system components from memory and process management to file system and system calls. Included building a fully-functional,

multitasking operating system.

Data Structures (OSU CIS214) – Subroutines and modular programming, searching, basic data structures, recursion, and an introduction to sequential files.

Professional Courses Taught

Intermediate Operating Systems (5-day course) – An introduction to working with operating systems (System V), building shell scripts, administration, and basic components.

Advanced Operating Systems (5-day course) – An extension to the intermediate class, going into detail about the inner workings of multitasking operating systems (using System V as an example). Discussions included advanced administration, devices, security, processes, and memory management.

Intermediate Java Programming (5-day course) – An incremental, highly focused course on the Java programming language that included most concepts of Core Java. Began with a section on proper object-oriented analysis & design. Concluded with building a server-based instant messenger and an RMI data-server.

C++ Programming (5-day course) – An introduction to C++ programming for C programmers which covered all common aspects of C++ programming (excluding the STL). Began with a section on proper object-oriented analysis & design.

Requirements Gathering and Analysis (2-day course) – A course that shows students how to collect and represent project requirements using a user-understandable methodology. Includes “feature walkthroughs” to ensure coverage.

Object-Oriented Analysis (2-day course) – A condensed course on taking documented requirements and creating the top-level object design using UML. Walkthroughs demonstrate compliance with the requirements.

Object-Oriented Design (2-day course) – A course that takes the Object Analysis and maps the objects to a specific language and filling in missing details (like containers). Walkthroughs demonstrate compliance with the requirements.

Professional Publications

Linux Socket Programming (SAMS/Addison-Wesley) – The experienced professional can use this book to write production-ready programs for the Internet and intranet (TCP/IP). The English version is out of print, but the Russian, Japanese, and Spanish versions are being sold.

Several Internet articles (InformIT.com) – Topics include security, Internet Protocols, peer-to-peer programming.

IPv6 Network Programming – A special edition of the *Linux Socket Programming* that shows the professional the techniques needed to write programs with the greatest longevity, using the next generation of the Internet Protocol (IPv6). [to be published]

Professional Reviews

The Ruby Way (Hal Fulton, SAMS) – Technical reviewer for the Ruby object-oriented language.

Inventions

“Language identification system and method for a peripheral unit” (US Patent #5,392,419)

Printer engine emulator for formatter development with proprietary RTOS.

Professional Memberships

American Computing Machinery (ACM, 1999)

Institute of Electrical and Electronics Engineers (IEEE, 2000)

Education

Computer Science Masters of Science (1990), Brigham Young University.

Computer Science Bachelors of Science (1988), Brigham Young University.