

2525 Roy Rd Unit 503  
Tyler, TX 75707  
[ranalger2000@yahoo.com](mailto:ranalger2000@yahoo.com)  
903-978-0167

# David Alger

## Objective:

To gain employment as a lecturer in the Computer Science department of the University of Texas at Tyler

---

## Relevant Experience:

2007-Current Tyler Junior College Tyler, TX

### **Professor / Coordinator / Department Chair, Computer Sciences and Visual Arts**

- Coordinator from 2017 – Current, Department Chair from 2017 – Current
- Courses Taught: Programming Fundamentals II, Introduction to Programming, Microcomputer Applications, Special Topics in Game Development, 3D Game Scripting I and II, 2D Game Scripting, Game Project Development I, II, III and IV, Mathematical Applications for Game Development, Introduction to Game Design and Development
- Taught courses in C/C++, Java, UnrealScript, TorqueScript
- Developed the Game Development AAS from nothing into a program that recruits about 80 students/year.

2001-2006 Business Knowledge Architects Roanoke, VA

### **Technical Director / Chief Software Architect**

- Directed the production of all software projects for the company in all phases of software development, including no less than five major projects for the Department of Defense.
- Sought out and won new contracts for the company, resulting in multiple new initiatives.
- Supervised, organized, and lead the software development team for the company.

1999-2001 Johns Hopkins University Applied Physics Lab Laurel, MD

### **Intelligent Autonomous System Designer**

- Designed and coded much of the Open Autonomous Kernel, an intelligent software package for autonomous control of US Navy ship systems.
- Published various internal and external papers detailing work on the Open Autonomous Kernel and Software Engineering.
- Presented information regarding Johns Hopkins system directly to Department of Defense customers.

1992-1999 United States Army Ft. Bragg, NC

### **Specialist, 98G Signal Intelligence Voice Interceptor**

- Maintained Top Secret/ SCI security clearance.
  - Trained and lead soldiers with valuable military equipment and classified information.
  - Assigned to Language Training Facility as an Assistant Language Instructor for Arabic.
-

**Education:**

**2000-2002**                      **Johns Hopkins University**                      **Baltimore, MD**

- Completed Master of Science in Computer Science with GPA of 4.0 with dual concentrations in theoretical computer science and artificial intelligence.
- Graded and tutored for Java Classes.

**1997-2000**                      **University of Maryland at College Park**                      **College Park, MD**

- Completed separate bachelor degrees in both Computer Science and Mathematics with 3.7 GPA.
  - Undergraduate Research in Graph Theory at Virginia Tech.
- 

**Academic Publications:**

**"A Distributed Model-based Control Architecture"**, The 6<sup>th</sup> World Multiconference on Systemics, Cybernetics, and Informatics, Volume VI, p.119.

**"Intelligent Control of Auxiliary Ship Systems"**, Fourteenth Conference on Innovative Applications of Artificial Intelligence, p.913.

**"A Distributed Model-based Control System for Engineering Plant Control"**, The Fifteenth International Conference on Systems Engineering (ICSEng 2002), August 2002.

**"Digital Archival of Technical Data – Get the Picture?"**, Proceedings, Undersea Defense Technology Conference, October 2004.

---

**Software Projects of Special Note:**

**Lexicon** – Developer, System automatically generated business rules based on near-human-language input from a non-technical user. Software integrated rudimentary natural language techniques, complex business rule processing, and scheduling techniques.

**Analysis of Waveforms for Emitter Identification** – Lead, Software analyzed incoming waveforms, condensed them using Fast Fourier Transforms, and compared the waveforms to previously collected repository to determine the source for the US Navy. Software integrated data warehousing techniques, numerical methods, and basic pattern matching.

**Self-Modifying Code for Java** – Lead, Software allowed technically trained users to change in-use business rules without exiting, recompiling, or reloading code. Development involved large-scale modification of large parts of the Java Runtime Engine and theoretical issues involving dynamic class medication.

**Fault Prognostication for Airborne Laser System** – Lead, Software using incoming data from onboard sensors and previous state information to predict faults in the chemical laser of the airborne laser system for the US Missile Defense Agency.

**Advanced Standard-Compliant Scheduling (ASCiS)** – Lead, Software allowed scheduling of arbitrary processes with dynamic selection of process parameters and settings. Scheduling engine was compliant

with RFC 2445. Large-scale project involving custom GUI features, time-slicing algorithms, multithreading conflict resolution, and state-determination reasoning algorithms.

**Jini-Enabled Business Rules Architecture (JEBRA)** – Lead, Company-altering paradigm for tactical business rule processing that included JTS, JTD, and JBR (outlined below). Architecture designed to define and execute complex processes in a tactical environment using oversight by intelligent software agents

**JEBRA Tactical System (JTS)** – Lead, Software package that was the tactical communication component of JEBRA. Software integrated intelligent software agent oversight, catastrophic failure recovery, data encryption, and automatic recovery. Project removed single points of failure, extended data bursts, intelligence vulnerability, and single node dependence from the communication system.

**Remote Java GUI** – Lead, Software allowed serialization of actual GUI components to remote terminals to obscure information being sent by the GUI, and allow remote terminals to have a zero-knowledge approach to transaction processing.

**JEBRA Transaction Dispatcher (JTD)** – Developer, Software extracted vital information from user interfaces, encrypted the information, and prepared transactions for processing under JBR (see below).

**JEBRA Business Rules (JBR)** – Developer, Software allowed dynamic definition of loading of complex business rules using well-defined meta-definition that could accurately contain any business process as described in business rules literature.

**Open Autonomous Kernel (OAK)** – Developer, Software used various intelligent components to control non-combatant on-board systems of naval vessels. Integrated model-based reasoning, trinary Boolean logic, advanced intelligent planning algorithms, basic autonomy algorithms, multi-agent architecture, and scheduling algorithms.