

*Background* I am interested in the fuzzy intersection of software engineering, programming languages, and computer systems. These three areas address how software is developed, expressed, and executed: three different perspectives on the central artifact at the core of computer science. While programming languages have traditionally been my primary concern, I am now increasingly involved in the other two areas as well. I have also recently become interested in the impact of information technology on society and policy.

*Approach* I approach research problems with an equal mix of theoretical analysis and applied experimentation. I strive to balance simplicity, efficiency, and safety, ensuring that my results remain accessible to a wide audience as well as applicable to realistic problems. I am equally comfortable working on my own or in collaboration with others; this includes working with graduate and undergraduate students as well as participating in inter-disciplinary research efforts.

*Experience* In recent years, my work in programming languages has focused on the design and implementation of component-oriented programming languages, resulting in the experimental programming language *Lagoona* which was the subject of my dissertation. I have also worked on the *transPROSE* project which explores novel ways to securely transport mobile code, focusing on inherently safe encodings orthogonal to cryptographic mechanisms. Most recently I have worked on improving public trust in electronic voting systems, specifically on an open computer-assisted approach to vote counting based on the Australian system.

*Plans* I plan to continue my work on programming languages and electronic voting in the future. For *Lagoona*, I specifically want to focus on flexible module systems that address deployment-side versioning as well as on efficient implementation of *Lagoona*'s generalized message dispatch mechanism. I also plan to work on software engineering issues in the coming years, focusing on agile development processes. Such processes replace detailed requirements analysis and system specification with informal meetings that produce "user stories" instead: concise descriptions of coherent pieces of functionality told from the client's perspective. User stories are customarily considered "less important" than other artifacts in agile processes, but if properly captured, user stories can provide valuable design rationales and navigation aids to mitigate the effect of personnel turnover and to establish "common ground" between agile and conventional processes.

*Funding* I worked with Prof. Dr. [Michael Franz](#) on grant proposal CCR-0105710 for the National Science Foundation. The proposal was funded with approximately \$180,000 from July 2001 to June 2004.