James Riely holds a PhD in Computer Science from the University of North Carolina at Chapel Hill, an MS in Computer Science from the University of California at Santa Barbara and a BS in Electrical Engineering from Northwestern University. He joined DePaul University's Foundations of Programming Languages group in 1999. Prior to that, he held postdoc positions at North Carolina State University and the University of Sussex and engineer positions at IBM and E-Systems (now Raytheon).

Research: Multicore processors, memory models, concurrent data structures and you.

Jane Cleland-Huang received the PhD degree in computer science from the University of Illinois at Chicago in 2002. She is currently a professor in the School of Computing at DePaul University, Chicago, where she serves as the director of the Systems and Requirements Engineering Center. She has served as Principle Investigator on grants worth $4 Million over the past 7 years. Dr. Cleland-Huang serves as Associate Editor for IEEE Transactions on Software Engineering, the Requirements Engineering Journal, and as requirements engineering columnist for the IEEE Software Engineering magazine.

Research: An Intelligent Approach to Certifying Safety-Critical Software-Intensive System

Research at CDM

Friday August 8th, 3:30 - 5:00 pm

Multicore processors, memory models, concurrent data structures and you.

Good programmers, like you, have a mental model of how their programs run. This informal semantic understanding allows you to predict what the program does. This mental model may be implicit and intuitive, much in the way our mental model of the physical world comes to us as children. But just as the childish notions of Newtonian mechanics break when things get small enough and fast enough, so does our mental model program execution. In this talk, I will reveal the awful truth about computers: the subatomic, timetravelling, mindbending weirdness of concurrent programming.

An Intelligent Approach to Certifying Safety-Critical Software-Intensive System

Did you ever wonder about the software running the airplane you flew in, or the software-controlled brake-assist system in your car? Like most people you probably assumed that the software would work correctly -- that the airplane wouldn't stop working midflight, or that your car wouldn't intermittently fail to brake upon demand. However – to achieve this degree of reliability, safety-critical systems need to be developed more rigorously than other kinds of software systems. In this talk I will present some of the work performed by my research group here at DePaul in building expert software traceability tools that support the development of highly fault tolerant, safety-critical systems.

Hosted by NSF REU Program
243 S. Wabash Ave. Chicago, IL 60604 CDM Room 723