Foto: Jessica Brandi Lifland/Polaris

Abstract from seminar by Edward A. Lee

Heterogeneous Actor Models

Complex systems demand diversity in the modeling mechanisms. We see this very clearly with cyber-physical systems (CPS), which combine computing and networking with physical dynamics, and hence require model combinations that integrate dynamics described using differential equations with models of software. We also see it in applications where timed interactions with components are combined with conventional algorithmic computations, such as in networked computer games. We even see it in traditional software systems when we have concurrent interactions between algorithmic components.

One way to deal with a diversity of requirements is to create very flexible modeling frameworks that can be adapted to cover the field of interest. The downside of this approach is a weakening of the semantics of the modeling frameworks that compromises interoperability, understandability, and analyzability of the models. An alternative approach is to embrace heterogeneity and to provide mechanisms for a diversity of models to interact.

In this talk, I will describe an approach that achieves such interaction between diverse models using a concept that we call "abstract semantics." An abstract semantics is a deliberately incomplete semantics that cannot by itself define a useful modeling framework. It instead focuses on the interactions between diverse models, reducing the nature of those interactions to a minimum that achieves a well-defined composition. I will illustrate how such an abstract semantics can handle many heterogeneous models that are built today (such as Statecharts, which combine state machines with synchronous concurrent models, hybrid systems, which combine state machines with differential equations, process networks, which combine imperative programs with message passing concurrency, etc.). I will also show how it handles combinations that are not readily available in modeling tools today. I will illustrate these combinations with examples prototyped in Ptolemy II.

About Professor Edward A. Lee

Edward A. Lee is the Robert S. Pepper Distinguished Professor in the Electrical Engineering and Computer Sciences (EECS) department at U.C. Berkeley. His research interests center on design, modeling, and analysis of embedded, real-time computational systems. He is a director of Chess, the Berkeley Center for Hybrid and Embedded Software Systems, and is the director of the Berkeley Ptolemy project. From 2005-2008, he served as chair of the EE Division and then chair of the EECS Department at UC Berkeley. He is co-author of nine books (counting second and third editions) and numerous papers. He has led the development of several influential open-source software packages, notably Ptolemy and its various spinoffs. He received the B.S. degree in Computer Science from Yale University, New Haven, CT, in 1979, the S.M. degree in EECS from the Massachusetts Institute of Technology (MIT), Cambridge, in 1981, and the Ph.D. degree in EECS from the University of California Berkeley, Berkeley, in 1986. From 1979 to 1982 he was a member of technical staff at Bell Telephone Laboratories in Holmdel, New Jersey, in the Advanced Data Communications Laboratory. He is a co-founder of BDTI, Inc., where he is currently a Senior Technical Advisor, and has consulted for a number of other companies. He is a Fellow of the IEEE, was an NSF Presidential Young Investigator, and won the 1997 Frederick Emmons Terman Award for Engineering Education.