6.897 Course Description

Basic information:
Time: TR 11:00-12:30
Room: 5-234

Prof. Nancy Lynch, NE43-365, 3-7225, lynch@theory.lcs.mit.edu
Prof. Daniel Jackson, NE43-530, 8-8471, dnj@lcs.mit.edu
Prof. Michael Ernst, NE43-524, 3-0945, mernst@lcs.mit.edu

Enrollment and prereqisites:
PhD and MS students in EECS; others (for example, MEng students in EECS and graduate students in other departments) with permission of one of the instructors. Students should have a strong background in mathematics, plus a good knowledge of computer systems (e.g., 6.033) and algorithms (e.g., 6.046).

Executive summary:
This course will explore the available state-machine-based methods for modeling and analyzing complex systems and will attempt to determine what is needed to make them more useful in practice. Students will study and learn to use a broad set of modeling techniques and analysis tools, and will apply them to case studies in areas such as data management in distributed networks, reliable network communication, mobile computing, automated transportation, and object-oriented programming.

Background:
Computer systems are becoming increasingly pervasive and increasingly complicated. The applications that are being constructed are becoming more ambitious (consider group games, support for military operations, electronic commerce, automated transportation, on-line data repositories,...). At the same time, the environments in which the systems must operate are becoming more dynamic (consider network changes, mobility, participants joining and leaving, failures,...). This combination leads to very complex systems, which can be hard to build and hard to understand.

One approach to making complex systems easier to build and understand is to develop precise system models and to try to use them in the system development process. These models can be used for documentation, for informal and formal arguments about correctness, and for performance analysis and prediction. Most of the popular models nowadays are based on some form of "state machines", which describe systems in terms of states and transitions.

Course overview:
In this new graduate course, we will explore the available state-machine-based methods for modeling and analyzing complex systems and will attempt to determine what is needed to make them more useful in practice.

We will study the mathematical foundations in depth, will learn the principles and usage of several support tools (model checker(s), theorem prover(s), an invariant generator,...), and will apply the mathematics and the tools to examples from several application domains. We will try to understand both the theory behind the models and methods, and how they can be used in practice.

The application domains currently being considered as focus areas are:

- Data management in distributed networks.
- Reliable network communication.
- Mobile computing (as in the Oxygen project).
- Automated transportation (e.g., air-traffic management or people movers).
- Object-oriented programming.

Each student should choose a particular focus area.

This course will have an interactive style—it won't be just a series of lectures. The class size should be small enough to allow everyone to participate actively. Students will be expected to read assigned papers and (parts of) books ahead of time, participate in discussions, and help to present some of the material. They will be expected to learn about, use, and teach others about, several tools, and to help develop a "library" of case study examples based on the application areas. Two kinds of examples will be developed: small toy examples to illustrate the use of the methods, and more elaborate examples that capture some of the interesting difficulties of the target application domains.

The class may split up into teams based on interest in the particular target application areas. The whole class will read and discuss the same papers and learn about the same tools. Different groups may apply the methods to different examples.

We will bring in some outside experts to participate in (or lead) some of the class meetings.

Partial course outline:
I. Simple Asynchronous Systems, Safety Properties
II. Simple Asynchronous Systems, Liveness Properties
III. Asynchronous Systems with Composition
IV. Timed Systems
V. Hybrid (Continuous/Discrete) Systems
VI. Probabilistic Systems

What students will actually do (for a grade):
This course will have no examinations or formal problem sets. Readings and modeling problems will be assigned; it is expected that everyone will make a serious effort to read and do as many of these as possible, within the 12 hours per week this course is supposed to take. Each student's grade will be based mainly on effort and participation, and in particular, on how well he/she helps others in the course to learn the material. As a concrete requirement, each student will hand in a "portfolio" at the end of the term containing the case study examples he/she worked out, and notes on any presentations or demos he/she prepared.

Collaboration policy: Anyone may collaborate with anyone on anything; in fact, collaboration is encouraged.