Com S 641
Semantic Models for Programming Languages
Spring 2004

Instructor

Dr. Markus Lumpe
113 Atanasoff Hall
294-2410
lumpe@cs.iastate.edu

Meeting Place and Times

TR 12:40-2, Pearson 213

Course Description

The catalog description of the course reads as follows:

Operational and other mathematical models of programming language semantics. Type systems and their soundness. Application of semantics to program correctness, language design and translation.

The main theme in Spring 2004 is “communicating and mobile systems”. Communication is a fundamental and integral part of computing, whether between different computers on a network, or between components within a single computer. Programs can be built from communicating parts, rather than adding communication as an extra level of activity. As a result, programs proceed by means of communication.

In Com S 641 Spring 2004, we will study programming language models for concurrent distributed interactive systems. The main focus will be on the \( \pi \)-calculus that provides a formal model of concurrent computation based upon the notion of naming, in which the topology of communication can evolve dynamically during evaluation.

The \( \pi \)-calculus differs from other models of communicating behavior mainly in its treatment of mobility. The movement of a piece of data inside a computer program is treated exactly the same way as the transfer of a message – or indeed an entire computer program – across the Internet. One can also describe networks, which reconfigure themselves.

The \( \pi \)-calculus is very simple but powerful. Its most prominent notion is that of a name, and it has two important ingredients: the concept of behavioral (or observational) equivalence, and the use of a new theory of types to classify patterns of interactive behavior. The Internet and its communication protocols fall within the scope of the
theory just as much as computer programs, data structures, algorithms, and programming languages.

**Textbooks**

There are no required textbooks for this class. However, you may have a look at the following recommended textbooks. If you have any questions concerning the textbooks (contents, should I buy it), please do not hesitate to contact me.

- Robin Milner, *Communicating and Mobile Systems: The π-Calculus*
- Davide Sangiorgi and David Walker, *The π-calculus: a Theory of Mobile Processes*
- Robin Milner, *Communication and Concurrency*
- Carl A. Gunter, *Semantics of Programming Languages*
- John C. Mitchell, *Foundations for Programming Languages*

**Prerequisites**

The formal prerequisite in the Iowa State catalog is successful completion of Com S 531 and Com S 541.

**Computer Accounts**

We will be using the computing facilities (Windows, Linux, and Solaris workstations) of the Department of Computer Science for all course-related assignments. The reference systems for all assignments are Pict 4.1 and Java 1.4.

If you do not already have a login on the departmental computer systems, you should get one (for details, please consult the department’s web page [http://www.cs.iastate.edu](http://www.cs.iastate.edu)).

The Computation Center and the Computer Science Department hold tutorials that are designed to help a new user to get familiar with their facilities. Please contact them for a schedule and information on signing up for one of these tutorials.

The course web page located at [http://www.cs.iastate.edu/~cs641/](http://www.cs.iastate.edu/~cs641/) will be used to post course materials including assignments, solutions to problem sets, lecture notes, etc.

As a user of the ISU computer facilities you have to familiarize yourself with the policies that apply (see [http://www.cs.iastate.edu/documents/policies.html](http://www.cs.iastate.edu/documents/policies.html)).

**Objective**

Com S 641 studies programming language semantics. In Spring 2004 the emphasis is on interactive systems. Interactive systems may behave differently, even though their
interaction sequences are “language-equivalent”. In fact, language-equivalence is not suitable for all purposes. For example, if we are interested in interactive behavior, then a non-deterministic automaton cannot correctly be equated behaviorally with a deterministic one.

At the end of Com S 641 students will have knowledge of, and be able to apply, the following aspects of the design, analysis, and implementation of interactive systems:

- What is strong bisimulation?
- What is weak bisimulation?
- How can observable equivalence be tested, used, and verified?
- How can interactive systems be represented in the π-calculus?
- What are the differences of synchronous and asynchronous variants of the π-calculus?
- What is sorting? How can sorting be used to classify patterns of interaction?
- How can we represent programming abstractions in the π-calculus?

**Assignments, Examinations, and Grading**

There will be regularly scheduled problem sets and programming assignments to help you learn the material and to allow us to evaluate your progress. Most assignments will require laboratory work. You should expect to work on a problem set between two and four hours. If you have trouble finding a solution, ask for help! All assignments are fair and reasonable. No problem set will require more than four hours. The necessary software packages will be provided on the department’s Windows workstations. However, you may need to configure them appropriately, before you can use them.

All problem sets are posted on the course web page. So, even if you have not received a homework handout, you can download the problem set from the course web page.

All problem sets are equally weighted. In general every problem set is worth a different number of points. The maximum number of points equals 100%.

Homework problems are due at the date and time specified on the handouts. In general, problem sets are in general due one just before the class starts. No credit will be assigned to late homework. If you have problems with a particular assignment, talk to the instructor before the deadline.

You have to submit your solutions on paper in class (preferable a printed copy of your code). You also need to submit your solution electronically. You need to logon to a CS department UNIX workstation and use the course's “turnin” script.

Your grade is independent of anyone else's grade in this class. That is, we do not grade on a curve, and everyone can get an A. Our purpose in grading is to uphold a standard of quality and to give you feedback: it is not to rank students.
Cheating

The simple rule of thumb is:

**Never give or use someone else's code or written answers.**

Such exchanges are definitely cheating and not cooperation. We will take action if we catch you cheating on a test or exchanging code or written answers. Read the section on *Academic Dishonesty* in the section on *Academic Regulations* in the *Iowa State University Bulletin*.

If you honestly believe that certain problems are too much busy work, then bring it to the instructor's attention; or failing that, only do the part of the problem that you think you need to do to learn the material and explain that to us.

Special Accommodation

If you have a documented disability and anticipate needing accommodations in this course, please make arrangements to meet with me soon. Please request that a Disability Resources staff sends a SAAR (Student Academic Accommodation Request) form verifying your disability and specifying the accommodation you will need.

Syllabus

The following gives a tentative list of topics not necessarily in the order in which they will be covered in the course. Specific reading assignments will be announced in class. Brief lecture outlines will be placed on the course homepage periodically.

- Interactive Systems
- Bisimulation
- The $\pi$-Calculus
- Early and Late Semantics of the $\pi$-Calculus
- Strong Equivalence, Observable Equivalence
- Asynchronous variants of the $\pi$-calculus
- Sorting and Typed $\pi$-calculus
- Interpretation of the $\lambda$-calculus
- Full Abstraction
- Encoding of programming abstractions