

Component Software & Technologies

Components: Where We Are, Why We Got Here and Where We Might Go

Shriram Krishnamurthi, Brown CS

Half the problem with a new technology is simply defining what it is and how it differs from what it isn't. Component technology is no stranger to this difficulty, or to the sibling perils of hype and abuse of terminology.

In this talk I give my views on what components are, how they are the consequence of an inevitable trend in software technology and user expectations, and what simple models we can use to reason about them. I also discuss some of the component-oriented research in progress at Brown and establish a context for the rest of the day's talks and discussions.

Optimizing Component Interaction

Mark Wegman, IBM, T.J. Watson

Optimizing component interaction is a fundamental problem in constructing efficient component-oriented programs. Our work reformulates the implementation of the interaction among components, with the goal of optimizing the assembled system. Our approach amounts to choosing automatically among a number of candidate implementation alternatives for each of the data structures and communication mechanisms used by the components during their interactions. Initial work has shown how to transform one example of component interaction optimization into a graph problem, and has proposed analysis of dynamics and fast graph-reduction heuristics as an efficient solution. We hope this work will inspire debate that ultimately will lead to a new area of investigation, within the field of compiler-performed optimization, with the potential to achieve order-of-magnitude improvements in the performance of component-oriented programs.

The Exploratories Project

Jean Laleuf, Brown CS Graphics Group

For seven years, the Exploratories project has been working on building exploratories—highly interactive microworlds for exploring domains—to teach computer science, mathematics and engineering. Advances in technology have facilitated the distribution and presentation of these materials, but the construction of the underlying software has remained a daunting task. The difficulties encountered are only compounded by the emergence of new domains and a rapidly changing workforce of undergraduate students. What we'd like—but have had trouble producing—are repositories of successful, reusable components to serve as building-blocks for future construction.

In this talk, I present our component framework for building educational microworlds. I briefly explain why this problem is hard and why existing frameworks aren't up to the task, but why we think that we've finally gotten it right. Our components have all been used to build microworlds for Brown's introductory computer graphics course and have recently been made publicly available for use at other institutions. Some of our microworlds will be on demo during the lunch break and late afternoon reception. Come by and play with them!

Better Separation of Crosscutting Concerns with Aspectual Components

Karl Lieberherr, Northeastern University

Proper separation of concerns is still of interest in software development. Here we focus on traversal-related concerns and how to encapsulate them as reusable components. Traversal-related concerns involve a group of connected objects that we traverse and manipulate to implement a behavior. Traversal-related concerns cut across multiple classes and are very common. Following the ideas of aspect-oriented programming (AOP), we would like to encapsulate a traversal-

related concern into an aspect. Doing this in the straightforward way leads to reduced scattering, but often results in an aspect that is tangled with accidental structural information. We notice a tradeoff between reducing scattering across classes and increasing tangling of structural information in the aspect.

Ideas from nondeterministic finite automata theory, however, can often simultaneously reduce both scattering and tangling. The well localized traversal-related aspects are conceptually aspectual components and can be simulated in plain Java using basically three classes from the Java package DJ. We explain the basics of DJ and how you can better structure your Java programs using ideas from aspectual components.

From libkern.a to COM to Knit: Three Generations of Component Models for Low-level Systems

Jay Lepreau, University of Utah

The OSKit is a collection of reusable components for building low-level systems. Experience shows that it provides a unique combination of practical power and principle. However, driven by users' experience with the OSKit, we've been forced radically to evolve its component model. The OSKit started as simple Unix libraries ("Genesis"), survived a dynamic object-based model ("Job"), and today offers special language support for static component definition and composition ("Revelation"). This talk will outline the journey and why Heaven still waits.

Lottery Enterprise Open Systems Initiative at GTECH

Mirek Kula, GTECH Corporation

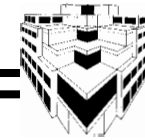
GTECH's new strategic direction is anchored in a number of macro-trends developing in information technology. These have mainly to do with the proliferation of Web-based standards, which have resulted in the wholesale disintermediation of entire families of economic agents and have empowered mass market consumers, as well as business entities, to engage directly in business and entertainment activities previously technically impossible or commercially unsustainable. This dramatically changing field has led GTECH to implement a new vision of the Transaction Processing Services enterprise. The presentation outlines the new company strategic vision, its business and technical manifestation and the experiences with technologies and methodologies that the company adopted in the process.

Building Components on the Network

Jim Waldo, Sun Microsystems

Component technologies have traditionally been targeted towards building applications that exist in a single address space. But enforcing such component models has been problematic, as programmers often trade off the isolation needed to comply fully with the component framework for gains in performance. The trend recently has been towards component frameworks in which network services are used as the building blocks of larger applications. Many of the component models that have been proposed have often been found difficult to change.

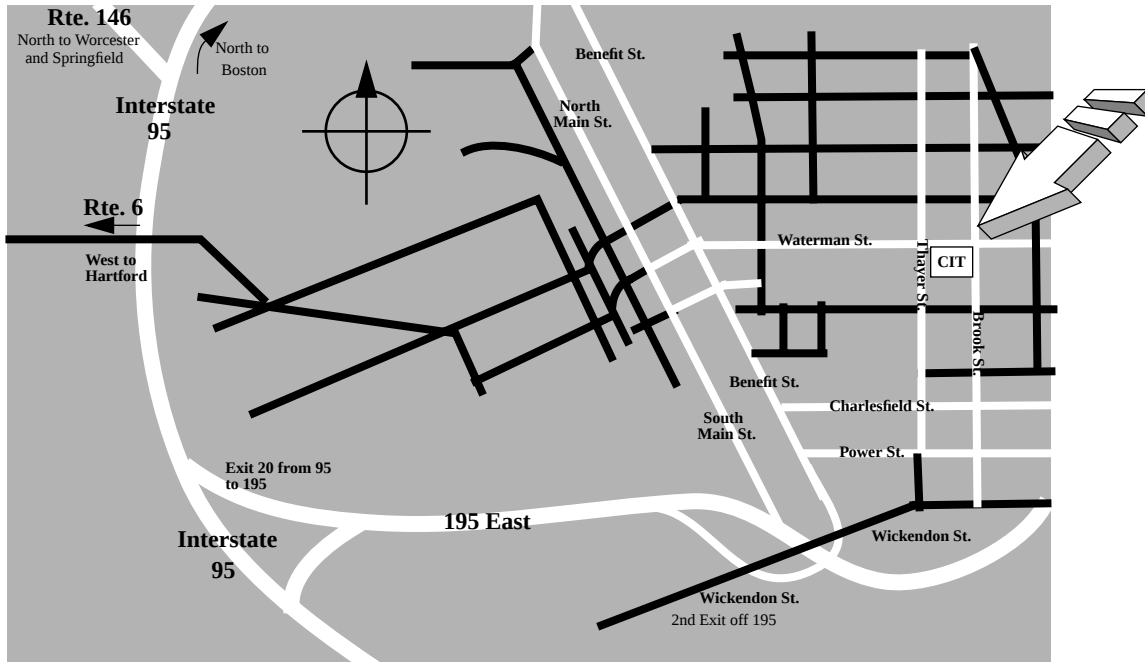
We discuss the Jini networking technology as a component framework based on mobile objects that allows the isolation of a network component system while allowing great flexibility in the evolution of the applications using the system. We show how violating some of the traditional assumptions of distributed component systems has enabled us to build a component framework that lets the network system evolve in interesting and important ways.



SCHEDULE

8:30	B R E A K F A S T and Registration 4th floor, CIT Building
9:15	Introduction and Overview Tom Dean, CS Chairman John Savage, IPP Director Shriram Krishnamurthi, Host
9:45	Components: Where We Are, Why We Got Here and Where We Might Go Shriram Krishnamurthi
10:45	Optimizing Component Interaction Mark Wegman, IBM Watson
11:45	The Exploratories Project Jean Laleuf, CS Graphics Group
12:00	B U F F E T L U N C H
1:30	Better Separation of Crosscutting Concerns with Aspectual Components Karl Lieberherr, Northeastern University
2:30	From libkern.a to COM to Knit: Three Generations of Component Models for Low-level Systems Jay Lepreau, University of Utah
3:30	B R E A K
3:45	Lottery Enterprise Open Systems Initiative at GTECH Mirek Kula, GTECH Corporation
4:30	Building Components on the Network Jim Waldo, Sun Microsystems
5:30	R E C E P T I O N (5th floor)





This symposium is a benefit of membership in our **Industrial Partners Program**. *Member companies are: Compaq, EMC, Foxboro, GTECH, IBM, InterTrust, Latitude, MERL, Microsoft and Sun.* There is no charge.

EMAIL REGISTRATION

To: sjh@cs.brown.edu

By: October 26

Please include the following:

Name

Title

Company

Department

Postal address

Phone/Fax

DIRECTIONS TO THE CIT BUILDING

- From I-95 N or S, take Exit 20 to I-195E.
- From I-195E take Exit 2, St.
- Go LEFT on Wickenden, LEFT again at the 2nd light onto Brook St.
- The red-brick CIT Building (Center for Information Technology) is on the left at the intersection of Brook and Waterman (1st light).
- Registration is on the 4th floor.

PARKING

Because most of the visitor parking has been assigned to University employees, I'm afraid we're unable to provide parking. Street parking is usually available for early birds, but watch out for newly designated 2- and 3-hour zones, which used to be all-day spots. You might try the residential area NW of the CIT.

The 28th IPP Symposium
Department of Computer
Science
BROWN UNIVERSITY

COMPONENT SOFTWARE & TECHNOLOGIES

Thursday
November 1, 2001

Hosted by Professor
Shriram Krishnamurthi

INDUSTRIAL
PARTNERS
PROGRAM

