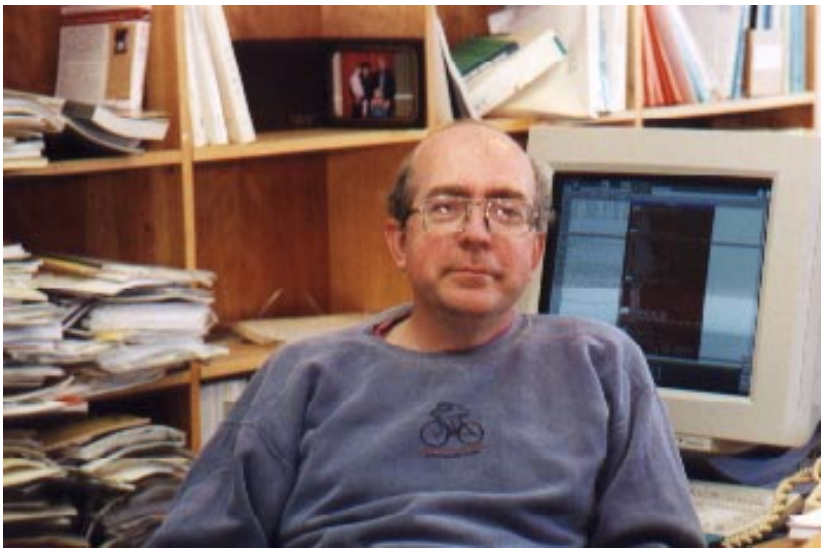
conduit!

Volume 7, Number 1 **Department of Computer Science** **Spring, 1998**
Brown University

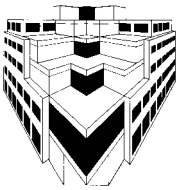
THE ALEPH TOOLKIT



Maurice Herlihy

I am currently working on the Aleph Toolkit, a collection of Java packages that implements an object-based, platform-independent distributed shared memory (DSM). The Aleph toolkit runs on any platform that supports JDK 1.1 or higher. This article describes the overall design and rationale for the Aleph API and internals, as well as our early experience implementing a DSM system in Java. The Aleph toolkit supports distributed computations running on networks of potentially heterogeneous processors. Aleph provides the ability to start threads on remote processors, to share objects among threads running on different processors (with transparent synchronization and caching), and to execute single-site transactions. A research goal of the project is to facilitate programming across heterogeneous architectures. Our emphasis is not merely on *portability*, i.e. adapting code written for one platform to another, but also on *interoperability*: the ability to run computations that span multiple hetero-geneous platforms. Our rationale begins with the following premise: the most effective way to support portability and interoperability is to program in Java to standard APIs, and to resist any temptation to tinker with the Java compiler or run-time system. Nevertheless, we recognize that on any particular platform, a customized native implementation is likely to outperform a portable Java implementation, no matter how well-designed. In response to this apparent dilemma, the design of the Aleph toolkit takes pains to isolate two performance-critical components of the Aleph run-time system: communication and data coherence. Out-of-box, the Aleph toolkit provides *default* implementations of the communication and memory coherence packages. Within the toolkit, each of these packages is accessible only through a Java *interface*, a language construct that constrains method signatures (and indirectly functionality). This approach makes it easier to replace the default package implementations with ‘‘native methods’’ (in practice, C programs) able to exploit specialized hardware devices (such as

> *“The most effective way to support portability and interoperability is to program in Java to standard APIs, and to resist any temptation to tinker with the Java compiler or run-time system”*



is also easier to experiment with application-specific data-coherence algorithms. Although the motivation for choosing Java may appear obvious, it still deserves some comment. JDK 1.1 includes a rich API: we saved enormous effort by exploiting the networking, reflection, and GUI packages. In the long term, we are interested in investigating topics such as check-pointing and process mi-gration (not currently supported). These problems are essentially impossible in languages such as C or C++, simply because the run-time system has no way to identify where objects begin and end. In Java, however, every object is self-identifying. Finally, although we have en-countered remarkable variations in the performance and reliability of different vendors' Java implementations, we are confident that these implementations will improve, thus indirectly improving our own performance.

The Aleph API

A toolkit client is a distributed program that runs on a number of logical processors, called *Processing Elements* (PEs). Each PE is a Java Virtual Machine, with its own address space. PEs communicate in two ways: one PE can start a remote thread at another, and PEs can share *global objects*. We now present a high-level description of the `Aleph` package.

Remote Threads

Perhaps the most economical way to describe remote threads is to review the traditional "hello world" application. The remote thread that will execute at each PE is defined as:

```
static class HelloThread extends
    aleph.RemoteThread {
  public void run() {
  System.out.println("Hello World
    from " + Aleph.thisPE());
  }
}
```

Remote threads must extend the abstract class `RemoteThread`. Like regular Java threads, the class must provide a public `void run()` method to be called when the thread is started. The `Aleph` class provides miscellaneous information about the run-time environment. Here, `Aleph.thisPE( )` returns the PE where the program is running. As usual for Java programs, the top-level class must include a method with signature

```
public static void
   main(String[] args)
```

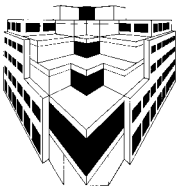
to be called when the program starts. The main method creates an instance of a remote thread object.

```
HelloThread thread =
    new HelloThread();
```

As with regular threads, a remote thread does not execute until it is explicitly started. The `main` method then creates a `Join` object for synchronization, enumerates all PEs, starts an instance of `HelloThread` at each PE (because starting a `RemoteThread` actually starts a copy of the thread object, a single `RemoteThread` instance can be started more than once), and waits until all remote threads have completed.

```
Join join = new Join();
for (Enumeration e =
     Aleph.allPEs();
```

President E. Gordon Gee enjoying Leslie Kaelbling's CS 148 class midterm (Building Intelligent Robots). The robots had to enter and exit the cul-de-sac.



```
        e.hasMoreElements();)
    thread.start
        ((PE))e.nextElement(),join);
  join.waitFor();
```

A *global object* is a container for a regular Java object. Here is how to create a global object containing a queue:

```
GlobalObject g = new
     GlobalObject(new Queue());
```

The method

```
Object open(String mode)
```

grants access to the contained object. For example,

```
Queue q = (Queue) g.open("w");
```

acquires exclusive access to the queue object contained in `g`. The method

```
void release() throws
      AlephException
```

indicates that the calling thread is done with the object. Under some circumstances, the release may fail, throwing `AlephException`. Our coherence model for global objects is based on *transactional memory*: an access is *successful* if the subsequent `release` does not throw an exception. All successful accesses are sequentially consistent, but unsuccessful updates have no effect and unsuccessful reads may return incorrect values. For example, here is how to update a shared queue:

"We feel it is essential that the run-time system should be able to 'go native' without requiring changes to application programs"

```
try {
   Queue q = (Queue) g.open("w");
   q.enq(x);
   g.release();
   } catch (AlephException e) {
      System.err.println("enqueue
           failed!");
     }
   }
```

The default transaction manager recognizes four modes: "`r`", "`w`", "`or`", and "`ow`". The first two denote *read* and *write* access, which have the obvious meanings and for which `release` never throws an exception. The last two denote *optimistic* read and write access, for which `release` might throw an exception. If an optimistic read or write is successful, then it has the same effect as its non-optimistic counterpart. An unsuccessful optimistic write has no effect, and an unsuccessful optimistic read may not be sequentially consistent. The optimistic modes are intended for objects for which concurrent method invocations are infrequent. Transaction manager implementations are free to support other modes. Any unrecognized mode is treated as a *write*. The Aleph toolkit also supports *transactions*, analogous to synchronization objects in Midway. Using a transaction, a thread can execute a critical section in which it atomically makes multiple updates to one or more objects.

Serialization and its Discontents

Message-passing among PEs is likely to be one of the more performance-critical aspects of the toolkit, but also the most amenable to improvement by native methods. For example, consider a cluster in which nodes communicate by a Memory Channel, or a collection of workstations linked by an ATM network. The only plausible way to exploit such resources is to call a ''native'' C program that in turn calls the specialized libraries needed by these devices. We feel it is essential that the run-time system should be able to ''go native'' without requiring changes to application programs. To this end, all low-level communication within the Aleph toolkit implementation is mediated by a *Communication Manager* interface. JDK 1.1 includes two features that appear well-suited to a DSM implementation: (1) object serialization, which allows an object to be written to and read from a stream, and (2) Remote Method Invocation (RMI), a remote procedure call package. We found object serialization essential, but RMI was not well suited to our goals. Instead, the default communication manager implementation uses a reliable datagram protocol for transport. We chose datagrams over stream sockets because most objects fit in a single datagram and lost packets are infrequent. An obvious way to transmit an object is to use serialization to write it to a `ByteArrayOutputStream`, convert that stream to a byte array, and create a datagram from that byte array. Unfortunately, stream-to-array conversion introduces superfluous copying and synchronization. To avoid this penalty, we exploited the Java class-inheritance mechanism by defining a `PacketOutputStream` class, extending `java.io.OutputStream` by overriding its `write` methods, forcing the object serialization methods to write directly to the datagram's byte array without further copying or synchronization.

On Graduate Recruitment Day—an opportunity for prospective graduate students to get a taste of life in the department. Graduate student Laurent Michel demos Numerica, a system for global optimization described in a recent book co-authored by Pascal Van Hentenryck and Laurent

Data Management and Coherence

A *transaction manager* package manages data coherence for global objects. To facilitate custom transaction manager implementations, all interaction between the transaction manager and other Aleph components goes through an interface. The default transaction manager implementation currently uses a simple invalidation-based, write-back caching scheme to manage global objects. The Aleph run-time system uses a bewildering variety of messages for global object and remote thread synchronization.

Ideally, we would like to mimic the behavior of *active messages*, encompassing a procedure's address and its arguments. When the message is received, the arguments are unmarshalled and control jumps directly to the procedure. Unfortunately, it is not possible to implement active messages as such in Java, but we can do something similar. We define an abstract class `Message` that implements `Serializable` and `Runnable`. This class provides methods for sending and receiving messages (via the communication manager). To define a new message type, one defines a class that extends `Message` by providing data fields and a `void run()` method that acts as a handler. Each PE has the same classes loaded at run time, so when a `Message` object is delivered to a PE, the PE deserializes the object's data fields and starts a thread to execute its run method. (Note that only the message's fields are transmitted, not its methods.) This use of class inheritance permits the PE's ''listen-loop'' thread to be implemented independently of the specific messages used by the Transaction Manager package.

Status

A preliminary version of the Aleph toolkit is up and running. It has been tested on Digital Unix, Sun Solaris, and Windows 95. We recently ran a distributed computation in which one PE was a Windows 95 laptop and the others Sun Solaris workstations. Porting Aleph to a new platform requires changing a few lines in a single file (typically just the pathname of the Java interpreter) and recompiling. The run-time system is structured as follows. Upon starting the toolkit, the user is presented with a console window. This window is used to start computations, to choose the platforms on which the computation will run, to view output, to control the computation's verbosity level, and so on. At each node on which Aleph runs, a *server* process listens on a well-known port. When the server receives a request to start a PE, it `execs` a process. More information on the Aleph toolkit can be found at *http://www.cs.brown.edu/~mph/aleph*.

Eugene Charniak

BAD TRIPS

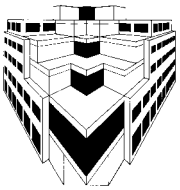
I remember being told once that at any first-rate university, about ten percent of the faculty are on planes at any given time. The import, of course, is that they are on their way to give talks, attend conferences, collect frequent-flier miles, etc. It occurred to me that some proportion of these trips must go awry in some way or another, and that almost every faculty member must have some not-so-fond memory that might in retrospect be amusing, or at least interesting. So I went to the faculty and asked. Actually, I have only asked about half the faculty at this point, but the stories were so good that I decided to go with what I have. Perhaps I will talk to the rest of the faculty for a second installment.

Fortunately, most of the stories indeed were amusing, though not all. The exception was John Savage's worst trip. In 1974 John was on sabbatical in Europe. On one trip to give a few talks his family came along. The first stop was in Bergen, Norway, after which they were scheduled to go on to Oslo. Unfortunately, John's son Kevin came down with an ear infection. The combination of a scared child and parents trying to deal with a foreign health-care system made for a situation that any parent can relate to. I was initially horrified to learn that in order to take the flight to Oslo Kevin's ear had to be punctured. John relieved my ignorant worrying by assuring me that this sort of puncture routinely heals up.

Interestingly, one trip story involved no travel at all. Peter Wegner was invited to give a talk at Columbia. In fact, he was invited three times. Unfortunately, this was during the 1968-69 school year and each time his talk had to be canceled because of the student uprisings. He never got there to give the talk (though he did give a talk there several years later).

As for travel problems, we rounded up the usual suspects. Both Steve Reiss and I were hit by snow storms in Rochester, NY. Steve had to walk a mile through his, I just got stuck in in Rochester for two days (it was the blizzard of '78).

Several of the faculty almost missed giving their talks. John Hughes tells of a trip from Providence to Purdue: "I caught an early-evening flight to Detroit, but there was rotten weather in Providence and we took off late. Then we circled Detroit for a bit while they cleared the runway—the weather there was worse. We landed in Detroit about 45 minutes late; I had a 45-minute connection. When I got off the plane, the "Departures" sign indicated that my flight was scheduled to leave at 8:30 pm. It was about 8:25 ... and the gate was at the other end of the airport. I ran like mad, finally getting there at about 8:33, only to discover that not only was my flight not leaving at 8:30, the aircraft wasn't even in Detroit. About three hours later, the aircraft finally arrived, and by 1:30 am we were aboard. Then we waited in the de-icing line for a while, and by about 2:15 we were in the air ... headed for Benton Harbor. We stopped there, discharged two of the other four passengers (this was a *small* plane) and



then, after some delays, took off for Purdue airport. We got there about about 4:30 am and, as it happened, the flight crew stayed in the same small on-campus hotel I was headed to, so there was a car waiting ... but I had to wait for the flight crew to come out. I got checked in to my room by about 5:15, and then (of course) couldn't fall asleep. I slept from 6 to 7, and then started my day.''

Tom Dean and Leslie Kaelbling teamed up for a good trip story. They were to report on some joint work at a workshop in Scotland called by the sponsors of the research. They agreed that Leslie would go, but a day or two before her departure she realized that her passport would be expiring fairly soon and that according to regulations she should not be allowed to take off. Hoping that she could bluff her way through, Leslie set off to the airport, after notifying Tom that he might have to pick up the slack. No go with the bluffing. So at this point Tom got tickets and set off himself, arriving at the workshop site about an hour before his talk. But his problems were still ahead of him. For starters, the slides for the talk had been prepared by Leslie, and as anyone knows who has tried to do it, talking from someone else's slides is very difficult. Fortunately, if that's the word, the slides were mostly useless anyway: Steve Hanks, a grad student with Tom at Yale and now a professor at the University of Washington, sidled up to him and said that the sponsors had decided that they were not interested in hearing about the project anyway and had given Steve a very hard time. So Tom threw away the talk and just kept the last two slides on future work. He managed to get through it somehow, stayed another hour, and then back to the States.

"My theory (which, thanks to this trip, I no longer adhere to) had always been that when you fly to Europe, the first day is wasted anyhow, so you might as well spend it traveling"

Tom Doeppner also had a classic travel experience. ''This one is more in the category of 'self-inflicted wounds': In September '93, I was invited to give a series of lectures at the CERN School of Computing, which was taking place in L'Aquila, Italy, 'just outside' of Rome. I figured this would be a good opportunity to get some more frequent-flier miles. So I checked, and yes, my airline did fly to Italy. From Chicago. To Milan. My theory (which, thanks to this trip, I no longer adhere to) had always been that when you fly to Europe, the first day is wasted anyhow, so you might as well spend it traveling. I put the theory into practice. To make the Chicago-Milan flight, I left Providence at noon, getting me into Chicago at 2:00. The flight to Milan left Chicago four hours later, at 6:00. I arrived at Milan at 10 the next morning, there waiting two hours for the bus to the other Milan airport, where the next flight to Rome left from. I then waited another hour at this airport, caught the flight to Rome, then took the subway, changing once, to get to the bus station. I waited there for an hour or so (standing on the street, as there was no waiting room), and finally got on the bus for the two-hour ride to L'Aquila. I arrived there at 8:00 pm, dead tired, but it was just dinner-time and I was expected to come to dinner and make pleasant conversation. I got to bed at 10, 26 hours after leaving Providence, several thousand miles richer.''

My favorite transportation-problem story, however, was told me by Roberto Tamassia. Roberto gave a talk in Cassis, France. On the way home he took a cab to the train station. He paid the fare, but rather than taking off, the cab stayed, presumably (Roberto reasoned) waiting for another fare. This seemed odd, however, because the station looked pretty empty. Indeed, further inspection revealed that it was not only empty, it was closed. Finally Roberto found someone who told him that there was a train strike and nothing would be leaving that day. At this point Roberto realized who the cab was waiting for—him, for a $100 cab trip to the airport.

When I first had the idea for this article, I had intended to end it with a story of my own. When I was interviewing for jobs I had one of the worst restaurant meals of my life. I was taken to a place that was really just a hamburger joint, but one with fancy paneling on the wall. Being thrown off by the paneling, I ordered a steak (medium-rare), only to get a well-done, stringy piece of what I hoped was at least beef. Needless to say, one doesn't want to get a reputation as a trouble-maker, so I simply ate what few pieces of meat I could find. Then, obviously not having learned my lesson, I ordered what the menu called ''strawberry shortcake.'' What I got was a piece of pound cake covered by something like Coolwhip

Eugene and Stan

(though I am slandering Coolwhip here) and decorated with a single half strawberry.

What makes this particular story worth telling is the fact that the job interview was at Brown. (The restaurant, now deceased, was Spats.) I intended to end with it because, given the Brown connection, I figured that it would be hard to top. However, when I told it to Stan Zdonik, he managed to top me by a mile.

In one of his job interviews, he was particularly interested in talking with one of the professors there who he knew had gone to the same grad school. Unfortunately, the professor in question fell asleep about half-way through Stan's talk. Stan was devastated and figured that his chances for a job at that institution had just gone down the tubes. His spirits revived somewhat when later he had a one-to-one talk with the professor, who apologized, saying that he had a newborn son and had not had much sleep the previous night. And, oh yes, the professor in question was me.

LETTERS & ALUMNI EMAIL

TED CAMUS PhD '95

Thought I'd get you up-to-date with my life in the real world. I've left my postdoctoral position at NIST (with fond memories) and in September 1996 joined Sensar as its 15th employee. Sensar's one product is IrisIdent, a device for verifying a person's identity using their unique iris pattern. The actual "iriscode" is extracted with algorithms developed by John Daugman, a professor at Cambridge, and is licensed through IriScan. What Sensar brings to the equation is the ability to scan the iris unobtrusively, the critical element needed for a consumer-level product. Our first application is the elimination of PINs at ATM machines; in just a couple of seconds your eye is automagically scanned and matched with an enrollment iriscode. The system even works through glasses and contact lenses. In June we demonstrated an early version at CVPR'97 in San Juan (although that prototype did not work with glasses).

For most of this year I have had primary responsibility for the Wide Field of View, which provides the (X,Y,Z) coordinates of the subject's eyes in 3D space. This uses stereo to provide depth information and various facial feature detectors to find the (X,Y) of the eye position. Using these coordinates, a calibrated pan-tilt-zoom camera system acquires a hi-res image of the subject's eye and provides it to Daugman's algorithm to produce the subject's unique iriscode.

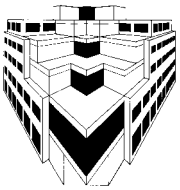
Sensar is making quite a bit of news recently, including the 12/15/97 ***Newsweek*** (you can find several AVI files at *www.sensar.com*). I am one of Sensar's two senior technical staff members (out of a current 54) and am actually the Acting Algorithms Director right now while my boss is on paternity leave. (I asked, but due to technicalities, I am not able to give myself a raise.) Sensar is a subsidiary of Sarnoff Corporation, whose consultants were critical in much of the early work on IrisIdent.

Tom—Congratulations on your new chairmanship! Eugene—Don't you think you're being a bit selfish with those Best Paper awards? Leslie—You know, I'll always regret not having time for your Lego robots course. Suzi—Thanks for the latest issue of ***conduit!*** I very much enjoy keeping up to date with the goings-on at the CS Department.

tedcamus@sensar.com

ANN CALDWELL, Former Vice President for Development

Dear Andy:... I am already planning to paraphrase something you once told me about the Brown Computer Science Department at my



new institution. You said "We don't prepare our students for the computer industry, we educate them to transform it." I promise attribution whenever appropriate and possible!

VICKI ESCALERA, University Auditor

Just a note to tell you how much I appreciate receiving the department's publication, ***conduit!***. Even when I don't have time to read it cover to cover or understand the background to each article, I still enjoy seeing the pictures of Computer Science faculty, students, staff, alumni and visitors. The tone of a vibrant department with individuals who enjoy their work and excel at it rings loud and clear.

KEITH LEWIS, friend of Professor John Hughes

A former colleague of mine from the Math Department now works at Bankers Trust. He sent me a long note, ending thus... "BTW, you're famous. One guy in my group was extremely impressed that I personally knew John Hughes, co-author of the most popular computer graphics textbook of the century. OK, your 15 minutes are up now!"

CHRIS NUZUM A.B. '92, Sc.M. '93

For those of you who remember me, don't worry: I haven't stopped stirring up trouble. In 1996, I left my job as Project Lead and Senior Software Engineer at Electronic Book Technologies to co-found Twisted Systems with Greg Lloyd '70.

After prospecting, product definition, and prototyping, we have a fantastic teamware product in development. "Traction" was inspired by generations of hypertext systems, drawing on elements of Augment, Xanadu and Vannevar Bush's visionary 1945 Memex proposal... It's also got a wonderful interface (designed by a RISD industrial designer), it's got Java brains, and is as easy as Quicken to set up and get hooked on. We're still looking for people to help make this venture a successs, so if this sort of thing excites you, please drop me a line! I'm still living in Providence, and, surprisingly, loving it. I spent four years playing electric bass in a local band (named successively Recurzion, Sharing Violation and Q) along with fellow Department alums Jeff Cutler-Stamm '93 and Mike Galvin '89. I also do a lot of photography and, recently, cinematography, and I'm looking forward to another summer of surfing, windsurfing and sailboat racing. But mostly I'm focused on making Twisted Systems a fabulous success. I love to hear from long-lost Brown friends, so please drop me a line. I promise I won't recruit you too hard, unless you ask me to. :-)

http://www.twisted-systems.com

MICHAEL RADWIN Sc.B. '97

Hi everyone: My Brown University honors thesis, "The Java Network File System," took 2nd place in the ACM "Quest for Java" 1997 Student Programming Contest. The contest was an international competition for Java applications and applets written by ACM student members. See *http://www.acm.org/jquest/webquest1.html* for a list of other winners. My project web page is at: *http://slimy.com/~mjr/projects/jnfs/*.

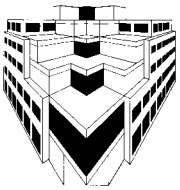
An excerpt from my paper: "The Java Network File Systems (JNFS) is a network file system for Network Computers (NCs). JNFS works on all "NC-compliant" NC devices, provides authentication and authorization support, works with other file systems such as NFS and NTFS and offers reasonable performance."

Thanks to everyone at Brown, especially Professor Doeppner, who served as my thesis advisor.

DAVID STEVENSON '96

Dear Andy, I just wanted to update you on where I've been since I TA'd CS92 for you in 1996. That fall I set off for the north woods of Minnesota to work in environmental education. It was a phenomenal place to spend the winter—snow stood four feet deep and temperatures regularly plunged to -20°F! I went there to be in the wilderness, but after nine months teaching middle-school students, I fell in love with the educational aspects.

This fall I followed my girlfriend to Paris where I found a school—the International School of Paris—that was very much in need of my services. They're heavily invested in technology and about to double their investment, yet are doing very little with it educationally. So, working with the headmaster and the teachers, I put together a report outlining the school's possible computing future. I thereby managed to craft for myself a Technology Coordinator job, working with the computers, students and, most of all, the teachers. It's great fun, it pays well, and I



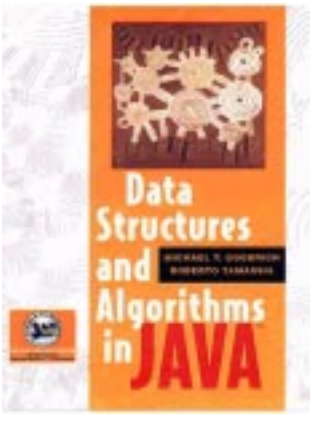
have a real chance to make a change here. Of course, when you come to a place with no email, any old way is up. I hope by spring to be able to point your CS92ers to student work on our website...

Anyhow, I thought you might be interested in an old CS92 TA who ended up in the profession. Thanks a lot for your vision and vigor. Thanks most of all for your high expectations—it took me a year 'out' to realize how rare and important that attitude is. I am in an exciting job in a spectacular city thanks in large part to the education I received from you and the rest of the CS department. My best to you, the department and the students.

d.stevenson@mailcity.com

TEACHING DATA STRUCTURES & ALGORITHMS IN THE JAVA AGE

Mike Goodrich



The design and analysis of efficient data structures and algorithms has long been recognized as a key subject in computer science programs. Indeed, the importance of data-structure design principles is actually increasing with the emergence of the object-oriented paradigm as the framework of choice for implementing robust and reusable software. Rather than simply viewing data as a collection of bytes and addresses, we think of data as instances of an *abstract data type* that includes a repertory of methods for performing operations on the data. Given the powerful influence of the object-oriented paradigm on the design of current software systems, we expect the central role of data-structure design in the computer science curriculum to continue and even increase in the years ahead.

A student in a data-structures and algorithms course should not only acquire a theoretical understanding of how a data structure is organized, but also learn the process of going from an abstract specification to a concrete running program. Textbooks on the subject typically include sample algorithms implementations in a specific programming language. Which programming language is best suited for teaching data structures and algorithms today? Many people believe that Java is the one.

Mike Goodrich at Johns Hopkins and I recently wrote a book on data structures and algorithms in Java. Our goal was not to produce yet another textbook on the subject, using Java code examples instead of C or C++ ones. Rather, we wanted to incorporate in our book a number of ideas we had developed through more than ten years of teaching. After selecting Wiley as our publisher, we started writing the book from scratch at the beginning of last year. About nine months later, the camera-ready manuscript was completed, and our book, entitled *Data Structures and Algorithms in Java*, appeared in the bookstores early this January. This article presents some of the main principles that guided us in writing this book.

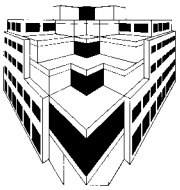
Visual Proofs

In this era of video games and *MTV*, students seem more visually oriented than ever. They learn most naturally by seeing a concept described with a picture, and they remember that concept by recalling the accompanying picture. This visual orientation is actually quite natural, since we humans devote an immense amount of brain power to processing visual information. Great educational benefits can be realized by finding visual ways of presenting the key ideas in important computer science concepts.

In a data-structures and algorithms course, many fundamental concepts are traditionally presented and justified by invoking sophisticated mathematical arguments. We have found, however, that one can often benefit from finding alternative visual explanations to replace or supplement mathematical proofs.

Consider, for example, the analysis of the *bottom-up heap construction*. A heap is a binary tree of logarithmic height that stores keys at its nodes such that the key of a node is greater than or equal to the key of its parent (see Fig. 1). Bottom-up heap construction starts by building one-key elementary sub-heaps. In the next step, pairs of elementary sub-heaps are merged together and a new key is added to

> *"We have found that one can often benefit from finding alternative visual explanations to replace or supplement mathematical proofs"*



form 3-key sub-heaps, and so on. The time spent by the algorithm to construct a sub-heap is proportional to the height of the sub-heap.

Bottom-up heap construction takes linear time. Traditionally, this result is proved by evaluating the following summation, which requires the application of various facts from calculus:

$$t(n) = \sum_{i=2}^{\log n} \frac{i}{2^i}(n+1)$$

We provide instead the visual justification shown in Fig. 1. We know that the work done to build a sub-heap is proportional to its height. We visualize this work by tracing a path down, starting at the root of the sub-heap. This path goes first to the right child and then to left children until the bottom of the sub-heap is reached. A simple inspection of Fig. 1 reveals that all the paths traced in this way are edge-disjoint, so that their total size is no more than the number of edges of the heap. Thus, the total work done by the algorithm is proportional to the number of edges of the heap, i.e., it is linear. Many other visual proofs are provided throughout the book. Our teaching experience confirms that students not only learn more easily with visual proofs, but also, and most importantly, they tend to remember longer those concepts that were presented visually.

While we strongly endorse the use of visual proofs, a caveat is in order. Pictures nurture and prompt discovery but they can sometimes be misleading. Mathematicians are especially aware of the pitfalls of fallacious demonstrations made via pictures that show specific instances of a problem and may not show special configurations where the property conjectured does not hold.

Design Patterns

Design patterns are a popular concept in the theory and practice of software engineering. A design pattern is a framework for solving a "typical" software design problem, a general template that can be specialized for the problem at hand. The reference book on the subject by the so-called "Gang of Four" (Gamma, Helm, Johnson, and Vlissides) describes and classifies 23 design patterns. It has been shown that design patterns save development time and yield software that is robust and reusable, and interest and excitement about design patterns continues to grow, both in academia and in the software industry.

In a typical computer science curriculum, design patterns are usually taught only in the software engineering course. However, important benefits can be achieved by using them as a pedagogical framework throughout the CS curriculum. At Brown, various design patterns (state, proxy, chain of responsibility, and factory) were being used by 1995 in teaching the introductory computer science course (CS 15).

We believe that design patterns fit in naturally with the discussion of several topics in a data-structures and algorithms course, and can be incorporated into such a course without major revision. Indeed, the design and implementation of many data structures and algorithms can benefit from the use of design patterns.

The design patterns used in our book are summarized in Table 1. The adapter, template method, decorator, and iterator pattern are also described in the book by Gamma *et al.*; the comparator, position, and locator pattern are new.

The *template method pattern* is especially impor-

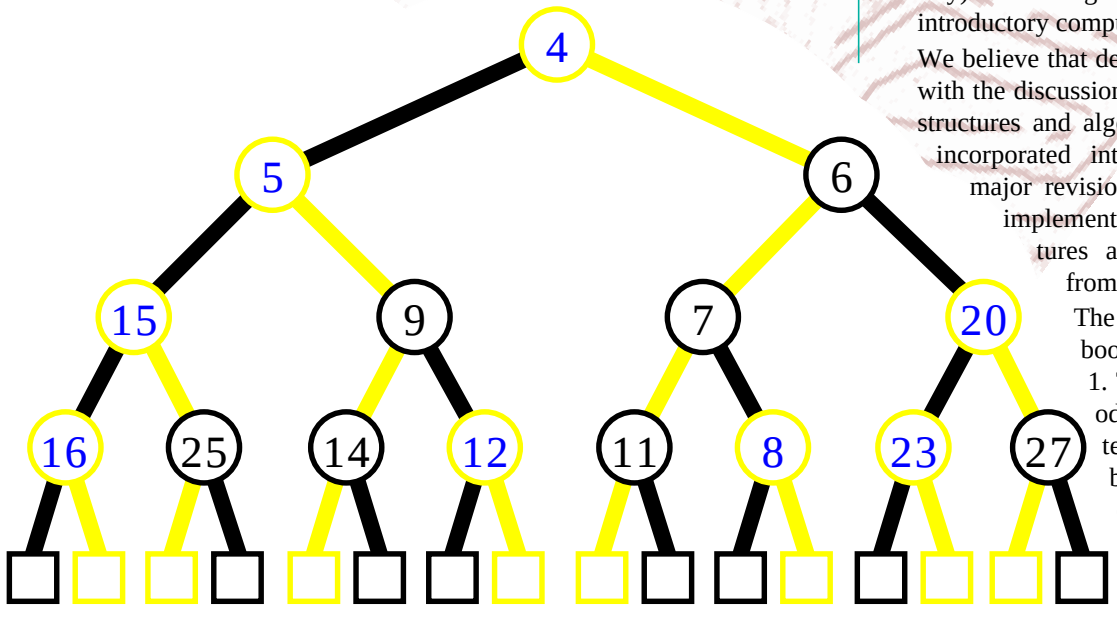


Figure 1: Visual justification of the linear running time of bottom-up heap construction, where the paths associated with the nodes have been highlighted with alternating colors—the path associated with the root of the heap consists of the nodes storing keys 4, 6, 7, and 11, plus an external node. Also, the path associated with the right child of the root consists of the internal nodes storing keys 6, 20, and 23, plus an external node.

l to r: Benoit Hudson, Luca Vismara, Mark Handy

Natasha Gelfand

adapter	stacks and queues
template method	tree and graph traversals
decorator	balanced trees, graph algorithms
iterator	sequences, trees, graphs
comparator	priority queues, sorting
position	sequences, trees, graphs
locator	priority queues, dictionaries

Table 1: Design patterns for teaching data structures and algorithms

tant for teaching algorithms. Several algorithms have the same overall structure but differ in the actions they take at specific steps. For example, many algorithms are based on a traversal of a tree or of a graph, but differ in the actions they perform when visiting a node. In such cases it is desirable to implement the core of the algorithm only once and then specialize it for the different applications. The template method pattern provides a class that implements a skeleton of an algorithm, and delegates to its subclasses the steps that will vary in different implementations.

Web Added-Value Education

Our book is accompanied by a continuously evolving Web site where we provide an extensive collection of educational aids, including:

- electronic slides suitable for class presentations;
- algorithm animations and interactive applets;
- source code for the data structures and algorithms in the book;
- testers and visualizers of student-written implementations;
- suites of test data for programming projects;
- additional exercises and project ideas;
- hyperlinks to other data structures and algorithms resources on the Web.

We view this web site as a ‘‘live’’ appendix to our book where we can share our latest ideas and teaching tools with colleagues and students.

Collaborators

A number of students at Brown and Hopkins have made specific contributions to the development of the book. Last summer, a team that included Ryan Baker, Natasha Gelfand, Mark Handy, Benoit Hudson, David Jackson, John Kloss, Amit Sobti, and Luca Vismara at Brown, and Edward Bross, Matthew Harris, and Russell Schwager at Hopkins assisted in the development of Java code examples and animations and in the review of drafts of book chapters.

Ryan Baker

The team, l to r: Matt Chotin, Ian Dembsky, Don Carney, Joe LaViola, Mike Demmer, Andrew Schulak, Ben Boer, Bob Zeleznik, Mike Coglianese, Dom Bhuphaibool. Not pictured: Saul Nadler, Saumil Doshi, Henry Tufo, Kerry

THERE AIN'T NO CS TEAM !

Coach Bob Zeleznik with Andrew "Shoe" Schulak (the author) in their natural habitat

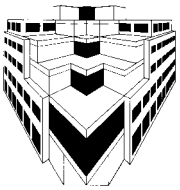
"...You must be joking." was the response of one player upon learning that the CS team was in first place.

How many of you know that the Computer Science Department sports teams play in Brown's intramural leagues?

And how many of you know that Bob Zeleznik has been involved in intramural sports for 14 years? Finally, how many of you know that last year CS won both the softball and football championships? Yeah, no kidding!

Before I describe our glorious football year, let me start off with the softball season, where it all began. Our team, the Dingers, had been playing okay ball all year with everybody hitting and fielding pretty well—we had won some tough ones and lost some easy ones—Steve Reiss was still putting artful spin on the ball while pitching, Tony Cassandra was playing in his last season and doing it like the fabled Cal Ripken, Jr., Andy "Fabio" Forsberg was playing with a recently recouped broken hand, I was hitting masterfully, Ben Boer was fielding tremendously and Joe LaViola was throwing bullets from the outfield. Individually we had a great team, the problem was getting the team together.

More junior players bemoaned "impossible assignments from malicious professors," while more senior players reported having to attend conferences where they were forced to be wined and dined in luxury by industry bigwigs. However, when a rainout fortuitously changed the date for our semi-final playoff game, we had players coming out of the woodwork—who wouldn't risk career and family for a chance at the championship in so prestigious a league? While much of the team had been playing for the Dingers for a good amount of time, there were some small personnel changes. One of them was the addition of rookie Robert Duvall. Fresh from an ankle injury from earlier CS intramural efforts, Robert led us off in the last inning with a brilliant hit resulting in a drive and culminating in a nine-run barrage. This sparked our advance into the championship game against the daunting Brown wrestling team the Ham'n'Eggers, aptly named it turned out. The championship game proved less dramatic. It was clear by the end of the first inning that we were destined for glory. As the runs piled up, the cheers for the Ham'n'Eggers diminished and lapsed into bickering. It was clear that they were fighting themselves more than us. Luckily we did not face such problems. The close-knit "family feeling" of the Dingers, bred from a shared heritage of all-nighters in CS labs, kept us



together, led us to victory, and more importantly, brought the T-shirts back home. Coach Bob credited our win to dedication and good play. I, however, am inclined to think the other team's beverage of choice may also have had something to do with it!

Riding high on our softball victory, we approached the football season. Last year our team, Public E---a, had made it to the championship game only to be slaughtered by a very tough team comprised of many members of the Brown rugby team. This year we would try to focus more on playing a defensive game. But again we needed players. We lost many of last year's players to contract negotiations, free agency and obscenely high-paying jobs in industry. We had to recruit, and recruit heavily we did.

Being a Head TA for CS15, I was in a good position to influence some of the TAs who worked for me. So I set myself out to fill their heads with visions of BBQ ribs, T-shirts and glory. But some people need more motivation than these obvious treasures. Some people need to see, to taste, to feel the gold that lies at the heart of a championship season. For these people I would aim to do no less. I told them I would fire them if they didn't sign up! So, with a full team of new recruits as well as some veterans, we attacked the league. We met all opponents as if they were the very devil himself and we were wandering apostles with buckets full of holy water to spread around. Despite our untried recruits and tired veterans, we had only one bad game during the season, which we lost by being too cocky. After this loss came some mental restructuring, and with our new hard-nosed attitude we marched into the championship game. With Bob as quarterback, myself on the line, and the rest of our merciless band of rogues and untouchables, we were the team to beat. But Bob would be away at a graphics conference for our championship game and he assigned the coaching job to me. I had coached a few of the Dingers softball games over the summer and had done okay (I had a 1-1 record as coach), but this was Public E---a football! This was serious! People's very lives depended upon this game!

"The close-knit 'family feeling' of the Dingers, bred from a shared heritage of all-nighters in CS labs, kept us together and led us to victory"

If you wish to be part of Brown's Computer Science intramural teams, please contact Bob Zeleznik or Andrew Schulak and they will be happy to give you some information and schedule a tryout for you.

We might all be geeks, but we're darn tough geeks. But for this game it wouldn't matter how much grace or grit we possessed, we still didn't have a quarterback, at least not a time-tested one like Bob. You see, Bob is so good at his position that he is able to execute the imaginative plays he calls with *style* and *panache*. But we didn't have that advantage now. It was time to buckle down and do some dirty blue-collar work. We were going to have to throw a rookie or two into the driver's seat of the out-of-control semi-truck that team Public E---a had become. This was going to be the most important game of our lives, and it did not look good.

Graduate students Joe LaViola and Don Carney got the call on game day. I started Don at quarterback then switched to Joe after half-time. Both Don and Joe threw for a touchdown apiece, and though our offense performed sub-par, it was good enough to beat the fraternity team Phi Kappa Psi. The real star of the game, however, was our defense. With junior Saul Nadler on the defensive line, the opposing quarterback faced a tireless rush the entire game. Saul chased him relentlessly and garnered a few sacks for himself as well as forcing a few key interceptions. When asked about his performance, Saul modestly said, "When I looked at him, all I saw was ribs. I didn't really know what I was doing, but I knew I wanted those ribs!"

Over the season, our defensive backfield sported such superstars as Ben Boer, Dom Bhuphaibool, Matt Chotin, Mike Coglianese, Ian Dembsky, Mike Demmer, Saumil Doshi, Kerry Kurian, Mike Legrand and Henry Tufo. Most players had to double on offense. But as Bob says, that's what makes us so tough: everyone is willing to play any position in order to get the job done. When the dust had settled and the wounded had been removed from the field, we were the victors with a score of *14-6!* All that remained was to email Bob the historic news. Why historic?—this win marked our first championship since the team was founded eight years ago.

On the season and the victory Bob said, "With all the big salaries and expensive perks our guys are getting, it's nice to know Public E---a can come out and play some hard-nosed, smash-mouth football when everything's on the line. Even the superstar head-cases like Saul were digging into the trenches just for the chance to wear the championship T-shirt and eat a pile of Wes's ribs."

Symposium speakers from l to r: Simson Garfinkel, freelance journalist; Richard Seltzer, Digital; Paul Kahn, Dynamic Diagrams; host John Hughes;

THE 20th IPP SYMPOSIUM

John Hughes

The 20th Industrial Partners Program technical symposium, held on November 10, 1997, was on "Web Technology: Security, Usability, Reliability, and Commerce," a topic selected for its relevance to the current interests of many Partners. The speakers included representatives from Partners IBM and Digital, Brown alumni from Dynamic Diagrams and INSO/EBT, and an independent journalist (who, I later learned, once applied to graduate school here), and their talks ranged from technology-user-oriented ("How can I make my website more secure? How can I draw more traffic? How can I make my website more effective?") to technology-producer-oriented ("What is needed for digital commerce to work? What is needed to make better web-content descriptions?")

The day began with a talk by **Richard Seltzer**, a self-described "Internet Evangelist" from Digital Equipment Corporation. His description of "The Social Web: From Hyper-links to People-links and Success on the Internet" characterized the Web as having changed from a world of "linking documents to documents and people to documents" to a world of "linking people to people." He described how his own Web page (freely supplied, purely text-based) had led to renewals of old friendships: the first thing people look for on the Web is themselves, and if they find you've mentioned them, they get in touch. From this, he made the additional point that the strategies of Web search engines should determine how one arranges the content of one's web pages—for example, the "title" field is one of the strongly indexed items. He also remarked on the folly of using images at a time when much Web access is through relatively slow modems, and therefore strongly advocated text-based Web pages. (In a department where multimedia notions have been standard for years, this is a provocative statement!) Through continued examples and anecdotes, Richard made everyone in the room want to check how well their own pages were indexed, whether they encouraged reader participation, whether they were indeed a part of the social web. I saw Prof. John Savage taking notes furiously during the talk: it was clear that he wanted to run to his office right away and start tinkering. One tidbit of note from this talk: since Alta Vista (and other search engines) are good at processing almost any text file, you can use them to index those mail files that you've been pack-ratting for lo these many years. Prof. Tom Doeppner, a legendary pack-rat of the department, was fascinated by this idea. If only there were some comparable way to index Prof. Andy van Dam's office...

The next talk, by **Steve DeRose** of INSO/EBT, was on XML, a proposed new standard for hypertext documents. XML is a powerful subset of SGML, which is a *metalanguage* in which "markup languages" can be defined. HTML, one such language, consists of a set of tags indicating various portions of a document (e.g., "Title," "Heading," "Subheading"). Unfortunately, these tags also include some formatting information; for instance, a 6th-level heading happens to generate small-caps formatting. That means that there are lots of Web documents that have "POSTSCRIPT" as a 6th-level heading, not because it's actually a heading for anything but

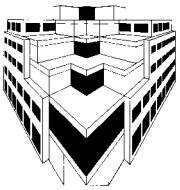
because the author wanted to format "PostScript" correctly. This and many similar problems lead to substantial difficulties in semantic indexing and have motivated the rethinking of HTML, leading to XML. XML lets one define one's own HTML-like language and then produce a document in that language (I like to think of an analogy with the "Style Sheets" provided by some word-processors). Steve noted that the major Web browser companies had signed on with support for XML in upcoming releases; he also predicted that certain new tagging languages, with tags appropriate to various applications, would become commonplace—average users wanting to make their personal Web page would not need to design a language first! (Indeed, they could still use HTML.) On the other hand, a company doing online commerce might well define a tag "PRICE" used to format all prices in its online catalog in the same recognizable way; such a tag would be unlikely to be defined in the "basic-Web-page-writing" language used by the common person. A key point was that hypertext documents should not be thought of as "formatting instructions" but as instances of a particular data structure—one defined by the tagging language—and that this makes them amenable to analysis via database techniques, in which the content and its presentation are distinguished.

Steve DeRose, INSO

Paul Kahn, President of Dynamic Diagrams, a Providence company specializing in information design, then spoke about the importance of creating an "information architecture" for a website, and how much good visual organization, as well as an understanding of the site-reader's needs, could inform this design. By good and bad examples, he showed how some commercial sites primarily provided a "corporate identity," with services to readers buried deep in the site if available at all, while others regarded their sites as opportunities to expand services to readers (FedEx's package-tracking is a particularly good example). Furthermore, the Web itself provides opportunities for creating new services by organizing available data—the various online weather-information services are good examples, as is the State of Washington's Seattle-Traffic-Viewing system: cameras and sensors located along major routes in Seattle show "what the traffic is like" with at most a five-minute delay. Kahn's talk provided an interesting counterpoint to Seltzer's earlier advocacy of text-only, reader-participant sites.

The next speaker was **Stu Feldman** of IBM's T.J. Watson Research Center, on business-to-business e-commerce. He began by carefully characterizing the difference between customer-to-business relationships and business-to-business relationships. Customer-to-business relationships are typically transient, involve little sharing of information (and what sharing there is is usually one-way) and individually not terribly important—the loss of a single customer does not greatly influence a business's success. In contrast, business-to-business relationships are typically long-term, involve substantial sharing of information that can be very sensitive, and may be the foundation of the business's success: the loss of its electricity supplier, for example, can ruin an aluminum refinery. These relationships put particular demands on any network used to support business-to-business e-commerce: security is essential, because of the information sharing, and reliability is critical as well—if one's bid on a new contract gets "lost in the ether," it can be a major financial disaster. Drawing on examples from IBM's experience in providing a reliable business-to-business information network, Dr. Feldman showed how these requirements would influence any successful design of a business-to-business e-commerce system.

The final speaker of the day was **Simson Garfinkel**, a freelance journalist who specializes in computer security and fraud. His talk described various kinds of security violations and how the infrastructure of computer networks—especially TCP/IP—leads to inherent security risks. He described various schemes for using other people's bandwidth (instead of storing a picture yourself on your web page, you invoke it from another's web page), sending IP packets as if you were at their computer (he mentioned that programs are widely available to do this—they "sniff" the ethernet and find out what sessions are active, and you simply click on a session to "take over" control of it!), grabbing passwords, etc. He also mentioned various ways to defend against these attacks, including the use of secure-shell technology for remote logins, disabling some critical features on web servers, using strong firewalls, and installing updates to software as soon as possible. He left the audience with the feeling, however, that this battle would never be won, just fought continually into the future. It was a sobering end to the day.



Eugene Charniak. Eugene is currently on sabbatical at Johns Hopkins University. He was awarded ONR funding to continue his research on "AI Approaches to Statistical Language Models."

Tom Dean. Besides his myriad duties as chairman and the various dynamic new initiatives he has launched within the department, Tom found time to give invited talks at AT&T Laboratories in New Jersey and at MIT.

Maurice Herlihy. Maurice has been named to the editorial board of the *Journal of the ACM*.

Philip Klein. Philip is a member of the program committee of the International Computing Combinatorics Conference. He and Leslie Kaelbling have been collaborating on two new experimental courses, CS17,18, which will offer potential concentrators a different route into the department than the long-standing, legendary CS 15,16 combination. Under the title "Computer Science: An Integrated Approach," the two courses will form a two-semester sequence. Stay tuned—we'll be giving you a full report after next semester's trial run.

Franco Preparata. Franco was invited to present a plenary lecture at the International Workshop on Algorithm Engineering in Venice.

John Savage. This past winter John completed his second year on Brown's Advisory Committee on University Planning as faculty Vice Chair. ACUP, a committee of students, faculty members and senior administrators, recommends a budget for the University to the President and the Corporation. This year, John and the student Vice Chair filed a minority report, the first in recent memory. Their report, which said that more should have been done to address non-competitive full professor salaries, generated a good deal of discussion on campus. They used the visibility of their report to raise other important issues including the high cost of a Brown education and the need for more financial aid.

Roberto Tamassia. Roberto co-chaired the program committee of the Workshop on Data Structures and Algorithms (Halifax) and served on the program committees of the Symposium on Graph Drawing (Rome) and of the International Colloquium on Automata, Languages and Programming (Bologna). He also gave an invited lecture at the Workshop on Geometric Computing in Nice. Together with colleagues at Johns Hopkins University, he was awarded a $1.25M research infrastructure grant from the NSF in support of research on geometric computing. His recent book on *Data Structures and Algorithms in Java* is described in this issue of ***conduit!***

Eli Upfal. Eli, our newest faculty member, gave a plenary talk at the 11th International Symposium on Fundamentals of Computing Theory (FCT) in Krakow, and he was an invited speaker at the Dagstuhl Seminar on Parallel and Distributed Computing in Germany. Eli is a program committee member for LATIN '98—International Symposium on Latin-American Theoretical Informatics—at which he will also present a paper. Said Eli, "And... most important, I moved to Brown in January!"

Andries van Dam. As director of The Graphics and Visualization Center (*http://www.cs.brown.edu/stc/home.html*), an NSF

Science and Technology Center, Andy guided the center through the review process to a successful renewal for another four years. The reviews of the work conducted in the center were exceptionally good and although the final award has not yet been determined, it will constitute a significant hike over the first four years of funding. Brown is one of five schools funded through this center.

Peter Wegner. Peter has assumed the editorship of a new Wiley series on computer science. He has been appointed for a second three-year term as editor of *Computing Surveys* and is continuing to participate in ACM's radical restructuring of publications from hard copy to electronic form. He is in his tenth year as editor of Brown's *Faculty Bulletin*, whose current issue focuses on faculty research and contains over 20 articles. This summer he will lecture in Spain and Germany.

ENGINE COMPANY 5 "ROUGH & READY"

Reprinted by kind permission of Tracie Sweeney, editor of the ***George Street Journal****.*

If you ask Mark Oribello how he spent a recent weekend, you'll swear by his matter-of-fact response that all he did was run a few errands around the town of Warren. And of course you'll be wrong.

Oribello, the media coordinator for the computer science graphics group, is also a volunteer firefighter/EMT in Warren. On the weekend in question, he and his colleagues on Engine 5—"Rough & Ready"—were called into action at two motor vehicle accidents and a chimney fire. A videotape he made at the scene of one accident gives a fuller picture of the weekend: sirens wailing, lights flashing, voices crackling over radios, the carcass of a vehicle that had hit a tree. Oribello got hooked on such sights and sounds while just a boy. "When I was seven, I watched every single episode of 'Emergency' on television," he said. In college, he obtained his emergency medical technician (EMT) certification, which required

Photo–John Forasté/Brown University

The Department's media coordinator for the graphics group, Mark Oribello, in his capacity as one of Warren's volunteer firefighter/EMTs

The Barco BARON, a large rear-projection display table, is a new addition to the graphics lab donated by Fraunhofer CRCG of Providence for use in collaborative projects. In contrast to the small viewing area of typical workstation monitors (see right of image), the Barco BARON affords a much larger viewing area better suited to group discussions and "semi-immersive" virtual reality applications. Current research projects include a 3D modeling system with a user interface that incorporates multiple modes of input and a VR minigolf game being created by the spring semester's VR-GISP group. In the 3D modeling system, arm and hand gestures are captured by electronic gloves and 3D trackers and are interpreted by the system to create new virtual objects and edit existing ones. Joe LaViola (l) and Andy Forsberg (r) illustrate how users might hold a design discussion. In the minigolf project, a real golf ball will be putted underneath the table and then a simulation of the results of that putt will be displayed on a virtual putting green

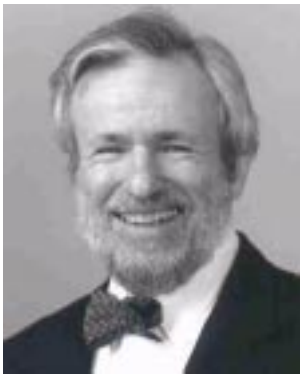
more than 100 hours of training, and written and practical examinations. Currently, he has extended EMT certification, which allows him to perform additional first aid procedures. He hopes to obtain cardiac EMT certification next.

Two years ago, Oribello became a part of Rough & Ready. "I'm one of the newer members," he said. "There are second- and third-generation firefighters in my company. Some of the others have stories that go back to World War II. ... The history of the engine makes people feel like they're part of something bigger." "We like the lights and the sirens," Oribello adds, "but we also get to be the good guys. Every member of the company, simply by being there when the call is put out, is doing something heroic."

Being a "good guy" means being on call. For Oribello, that has meant being interrupted while in the shower, eating dinner, or at 2 or 3 in the morning. He carries a pager with him at all times, and has emergency gear stashed in the trunk of his car. It's all "just part of the job," Oribello said. "I have a responsibility, as does everyone in the department, to serve." The responsibility sometimes means working in an adrenaline-charged atmosphere of shouting, anger, confusion and pain. "You never see [the victims] at their best in this job," Oribello said, "which sometimes means volunteers' efforts go unappreciated." But over the years, you see folks get praised. That really goes a long way."

Lisa Manekofsky, Administrative Manager for Andy van Dam's NSF Graphics & Visualization Center, was recently awarded a 'Brown Says Thank You' award for exemplary service to the University. President Gee gave the award a new twist by dropping by himself to announce it and offer congratulations. Lisa's mother (to Lisa's left) shared the surprise and joined the festivities

CHARNIAK UNPLUGGED

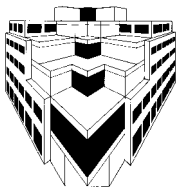


Eugene Charniak

I would like to start by disclaiming any responsibility for the title of this column, though when asked, I could not think of a better one. I would guess that our editor-in-chief is responsible, though it is possible that it is the mere editor's doing. I must say, though, that when I got my copy of the last issue of ***conduit!*** and turned to the back to make sure my name was spelled right, I laughed out loud. I assume that "unplugged" is meant to be a combination of "untethered" (from the chairmanship) and "unhinged."

I was standing next to a printer a few months ago when our facilities manager, Jeff Coady, stopped to ask me if my job came out OK. After I said ''yes,'' Jeff said they had been having some trouble with the printers: none of them would print out the drafts of John Savage's forthcoming book. I smiled and said that this was probably because John is a theoretical computer scientist. This left Jeff perplexed, so I explained that in physics there is a long-standing tradition that theoretical physicists should not be let near any experimental equipment: their mere presence is enough to screw everything up. The Nobel laureate physicist P. M. Dirac was said to be so gifted in this way that he had only to be in the same city for things to go kablooey. Jeff responded that the facilities group here once had an ''anti-Dirac:'' Dave Durfee just had to lay his hands on a piece of equipment and it would start working properly. Even now, when nothing else works, they try saying, ''Dave Durfee, Dave Durfee'' to see if it helps. (Of course, Dave went on to get his Ph.D. I forgot to ask Jeff if he thought that might have killed his abilities.)

Every Monday the ***New York Times*** business section is devoted to issues in information technology. A year or two ago it had an article about Eric Albert '80 and his crossword-puzzle software (***conduit!*** V. 5 No. 2). Last November Edwina Rissland '69 had her picture there in conjunction with an article on a new software product that helps with patent-infringement problems. Edwina is a professor of computer science at University of Massachusetts Amherst and president of the International Association for Artificial Intelligence and the Law. She is well known for her work in AI approaches to legal reasoning (I was aware of her work long before I learned she was a Brown alumna). Edwina served as an adviser and occasional consultant to the company, though the ***Times*** assures us she has no financial stake in the project. The accompanying picture of Edwina



was a nice one, and I asked Suzi Howe to see if the Times would give us permission to use it. They would, but only for $200. I didn't like it that much.

When I became chair of the department six and a half years ago, I was due a sabbatical that year. Sabbaticals at Brown are ''use 'em or lose 'em'' in that you do not start accruing credit for your next one until you have taken the one you are due. Exceptions are made, however, in exceptional cases, and clearly it made no sense to become chair and leave for a sabbatical. Thus when I retired from the chairmanship I had accumulated credit for two sabbaticals. I got the university to agree that if I used one now, I could save the other for a bit. Unfortunately, my son is in high school and neither he nor Lynette, my wife, looked very favorably on the idea of leaving Providence. On the other hand, Southwest Airlines has recently started flights from the new Providence airport and prices have really come down. In particular, flights to Baltimore are now very affordable, and it turns out that Johns Hopkins has several excellent faculty members who work in my research area (statistical natural-language processing). Thus we decided that I should spend my sabbatical at Johns Hopkins and return to Providence every weekend. The way it is working out is that I leave for Baltimore almost every Monday morning and return Thursday night; Fridays I spend at Brown meeting with my students. By and large this has been easier than I had expected. I now see how long-distance marriages are possible. As I told Lynette, though, the hard parts are Monday through Wednesday evenings. They are pretty lonesome. So Lynette bought me a jigsaw puzzle of a polar bear on snow. I am making slow progress.

conduit!

A publication of
The Computer Science Department
Brown University

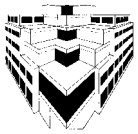
Inquiries to: ***conduit!***
Department of Computer Science
Box 1910, Brown University
Providence, RI 02912
FAX: 401-863-7657
PHONE: 401-863-7610
EMAIL: sjh@cs.brown.edu
WWW: http://www.cs.brown.edu/publications/conduit/

Suzi Howe
Editor-in-Chief

Katrina Avery
Editor

Jeff Coady
Technical Support

Department of Computer Science
Brown University
Box 1910, Providence, RI 02912, USA



conduit!

NON-PROFIT
U.S. Postage
PAID
Providence, RI
Permit #202

Printed on recyled paper

Address changes welcomed