stc.org/region8/nbc/www/

northbay news

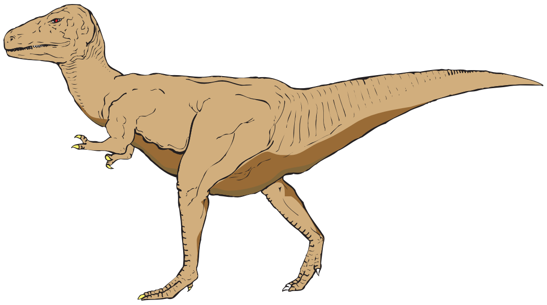
The monthly newsletter of the NorthBay Chapter of the Society for Technical Communication

Volume 7, Number 6, June 2000

In This Issue

May’s Meeting: Visit from PacificTec

John Dibs
President



“I would have preferred distinction.”

When the Sonoma County Water Agency needed a full-service documentation solutions provider, who did they turn to? As Chris Steele explained on May 18 to a near-record STC crowd, PacificTec Progressive Solutions, Inc. that’s who.

For the evening’s theme, *Distinct or Extinct: A Method of Success for the New Millennium*, Chris, Chief Operations Officer at PacificTec, shared the company’s values and highlighted their achievements. Founded in 1991, PacificTec leverages technical writing, technical illustration, and what they call new media skills to help clients solve problems. The firm meets this goal through three separate, inter-related divisions: Technical Publications, Custom Solutions, and Education & Training. STC

Continues on page 3 ☞

Writing for Programmers, Part 2

by Ken Delpit

The second part in a three-part series of articles about writing for programmers inspired by Andrea Ames’s presentation at the March North Bay STC meeting.

These days, the programming world has gone heavily to object-oriented programming (OOP). As technical writers, can we write about this subject without actual programming experience? As described in Part 1, Andrea assured us that we could.

In Part 2, we discuss the meaning of and explore the fundamental concepts of OOP, and compare it to traditional programming. In Part 3, the final installment, we will discuss the OOP application development process, compare various OOP environments, and reconsider the role of the OOP technical writer.

What Is Traditional Programming?

In order to understand the concepts of OOP and appreciate its benefits, it is helpful to consider non-OOP, that is, traditional programming. Programmers using traditional

Continues on page 3 ☞

northbay officers and committee chairs
john dibs
(707) 792-1791
jdibs@earthlink.net

vice-president (programs)
barbara herbert
barbara@sonic.net

first vice-president (programs)
kurt huget

newsletter
editor, john dibs
publisher: shelley hoose
(hoose@mac.com)
newsletter staff:
ken delpit (kdelpit@compuserve.com)
gabrielle de serres
(gabrielle.de.serres@usa.alcatel.com)
barbara herbert (barbara@sonic.net)
michael meyer(mmeyer@sonic.net)

web
shelley hoose

membership/telephone tree
whitney parker
(707) 537-1792
whitneyp@aol.com

hospitality
michael simoni & gabriella de serres

treasurer
carolyne gibson
(707) 577-2529
carolyne_gibson@hp.com

submitting articles and ads

We welcome articles, advertising, and news about meetings, workshops, and courses that pertain to technical communication. Please email simple text to the editor at jdibs@earthlink.net
For our current advertising rates, please email or phone the editor.

reprints and distribution

If you reprint articles from the *northbay news*, please credit them and forward a copy to the editor. Reprints in non-STC publications are subject to the author's approval. Copyright © 2000 *northbay news*. *northbay news* is free to NorthBay Chapter members. Nonmember subscriptions are $6 per year.

STC Mission Statement

The mission of the Society for Technical Communication is to improve the quality and effectiveness of technical communication for audiences worldwide.

This Month's Meeting

Thursday, June 15, 2000

"A Case Study in Usability"

Mic Vandersluis, Senior Member, NorthBay Chapter

This month, our stalwart meeting host discusses the practical issues of infusing usability awareness into the product development mindset. Mic's presentation covers these topics:

- What is usability and what are user-centered design (UCD) principles?
- Selling usability to the company
- Influencing product usability on the design team
- Testing methods
- Dedicating resources to usability assurance
- The proof is in the pudding: recognizing a value-added activity
- Resources for learning about usability
- Q&A

... and, you'll get to participate in real live usability tests. Cool!

A technical communicator for over thirteen years, Mic, like other seasoned tech-comm professionals, focuses much of his energy and career interests on product usability. Drawing on extensive experience as a user advocate on product design teams, he currently advocates usability awareness and implements user-centered design techniques at Compumotor, where he has worked for the past eleven years.

Join us as Mic relates his experiences as a usability evangelist and budding usability specialist on his design team at Compumotor.

Meeting Time & Schedule

Date:	Thursday, June 15	
Location::	Parker Compumotor, 5500 Labath Dr., Rohnert Park	
Time:	5:30–6:30	Networking, Show and Tell
	6:30–6:45	Introductions, Announcements
	6:45–7:45	Program
	8:00–8:30 . . .	More Conversation, Idea Swapping

May's Meeting

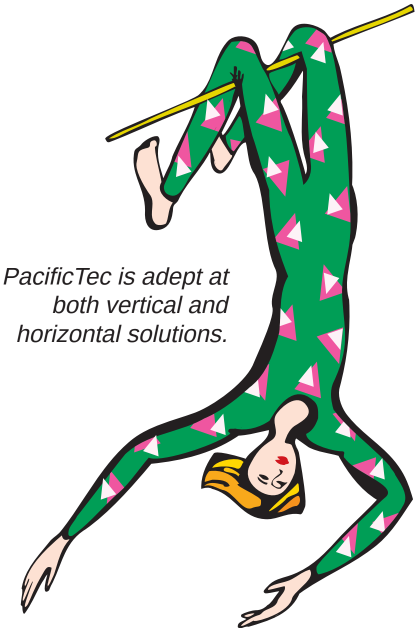
Continued from page 1

member Cyd Dunning heads up the Technical Publications division.

For PacificTec, being distinct means providing client solutions with both *horizontal* and *vertical* dimensions. The horizontal dimension involves meeting the needs of the task at hand — creating and maintaining up-to-date technical information and training material. The vertical dimension involves having expertise in a variety of industries.

A critical key to success is the ability to understand and adapt to the needs of various audiences and cultures. Each audience's culture, language, and value set play critical roles in deciding how to convey information. Technical communicators need to be fluent with their audiences in order to be distinct.

The Sonoma County Water Agency, featured in a public relations demo produced by PacificTec, had scarce, out-of-date, or inaccessible documentation.



PacificTec is adept at both vertical and horizontal solutions.

PacificTec solved the Agency's documentation need through a combination of dynamic CAD drawings, video clips, and standard operating and maintenance procedures, all delivered via a browser interface over the Agency's intranet for easy access. Chris displayed zoom-enabled, scaled plant drawings as the best format for meeting the Water Agency's documentation needs. Another solution, a video training component with narration, provides instructions for Agency employees on how to access the wastewater facilities at the Sonoma County Airport.

For Cyd Dunning, the company's location in Windsor, and the variety and opportunity at PacificTec keep her happy. She explained that PacificTec communicators wear lots of different hats, enabling them to work as freelancers without having the marketing aspect of the job. Russ Stansbury, also present on May 18, recently moved into a marketing role at PacificTec.

A critical key to success is the ability to understand and adapt to the needs of various audiences and cultures.

Several of our STC chapter members have interacted with PacificTec staff at places such as Hewlett-Packard/Agilent. PacificTec's desire to make their presence felt in the NorthBay STC was experienced in full force May 18. Visit **www.PacificTec.com** or call 707-838-0918 for more information or to inquire about employment opportunities Ask for Chris, Cyd, or Melanie in Human Resources.

Writing for Programmers

Continued from page 1

programming techniques tend to "think" and "see" in *verbs*. That is, they design their applications as systems of *procedures*, or transactions. They then develop programs that perform the procedures in particular sequences at specific times.

For example, an application built to process transactions for a public library might include programs to check books out, check books in, calculate late charges, place books on hold, notify borrowers that held books are awaiting pickup, and so on.

For the most part, traditional programming has served us well, and continues to be used in business. Though no longer in vogue, traditional systems will be around for years to come, and in fact, continue to be developed. Interestingly, many so-called legacy systems are quite modern in construction, employing modular design and well-crafted implementation.

However, from a developer's point of view, traditional designs present several problems, including difficulty and duration of development, and difficulty and duration of change. While many traditional systems have been built quickly and designed cleverly with change in mind, such systems, in general, resist both rapid development and rapid change.

What Is OOP?

In contrast to traditional programmers, OOP programmers tend to think and see in *nouns*. That is, they see and design their applications as systems of *objects*, on which functions are performed at various times, not necessarily sequentially.

An OOP library application, then, consists of objects, such as books, borrowers, notifications, and so on. Each object would contain, or *encapsulate*,

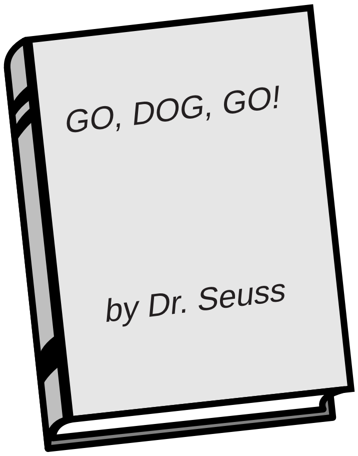
Continues ☞

Writing for Programmers

Continued from page 3

both data and functions. The data portion of an object’s definition describes attributes, or *properties*, of the object. The functions portion of an object’s definition describes operations, or *methods*, that the object can perform on itself.

In the OOP library application, the book object would probably encapsulate properties of book title, author name, library catalog number, publisher name, publication date, number of pages, and so on. The book object would encapsulate methods suitable for books, such as those for checking itself in, checking itself out, calculating late charges, and so on.



An example of an object and its main method.

Application programs, then, would invoke the appropriate methods at appropriate times by sending *messages* to the objects. Messages tell the object what to do, by referencing the method name, and pass variable data, or *parameters*, to the method. The syntax, or *signature*, of a method is specified by the developer, according to the development language, and must be adhered to in execution. Generally, a method-invocation message consists of a method name and a list of parameters, some of which are mandatory and some of which are optional. For example, the signature of a C++ function that calculates late charges on a book might be:

```
double CalcLateChg
(int NumDaysOverdue = 0,
double DailyRate = .25);
```

This signature specifies:

- the name (“CalcLateChg”) of the method and the type (“double,” or double precision) of the method’s return value
- the name (“NumDaysOverdue”) and type (“int,” or integer) of the first parameter, and the default value of the parameter (“0” days) if not otherwise specified
- the name (“DailyRate”) and type (“double”) of the second parameter, and the default value of the parameter (“.25” dollars, or 25 cents) if not otherwise specified
- a C++ statement terminator {“;”)

Signatures are good things for OOP writers to get to know. Collectively, they form a major part of the application programming interface (API) for a system. Signatures tell the programmer how to invoke the system’s methods. In general, the API tells the programmer what he or she needs to know in order to develop applications that use the system’s objects.

Developing API documentation is, therefore, a common request made of writers. But, take heart. The API is just an interface. By itself, an interface conveys nothing about a system’s inner workings. The programming behind an interface may consist of a few lines of code or it may consist of thousands, but the beauty is, you don’t care! By learning a few syntax rules and programming conventions, you’ll be able to “read” code well enough to document an API.

What Are We, Verbs or Nouns?

When you think about it, the noun (object-oriented) perception more closely mimics the real world than does the verb (traditional) view. As David A. Taylor says in *Object Technology: A Manager’s Guide*, “Why look for some other way to package procedures and data when the real world has already organized them for you?”

Indeed, after creating the first book object, an OOP developer would soon discover that all book objects need to have (encapsulate) these same characteristics (properties and methods). Rather than redefining characteristics for each book in the library, the programmer uses the tools provided in the OOP environment to set up collections of similar objects, or *classes*. In this way, the programmer defines a generic book object just once, and can create specific *instances* of books as often as needed. An object, then, is simply a particular instance of a class.

Moby Dick, *Catcher in the Rye*, and *Managing People for the Complete Idiot* (real title; if your boss has this one, draw your own conclusion) would be individual instances of class “book.” Each would have unique data associated with it (title, number of pages, and so on). But what about special objects that require unique characteristics? For example, volumes of an encyclopedia would need a “volume number” property that has not been defined for the generic book class.

No problemo. The programmer need only *instantiate* (that is, create an instance of) class “book,” and assign an additional property “volume number” to that object. Objects *inherit* all characteristics of the class from which they were instantiated, and they have any additional properties and methods defined specifically for that instance.

But there’s more. Just as it made sense in our library application to define a generic book class, it makes sense to define a particular type of book *subclass*, such as “reference.” The “reference” subclass inherits all properties and methods defined for class “book,” and has unique characteristics associated with itself. So, now an object instantiated from

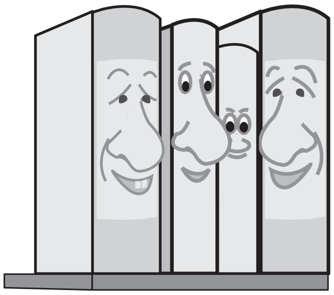
Continues ☞

Writing for Programmers

Continued from page 4

subclass "reference" inherits all properties and methods associated with both class "book" and with subclass "reference."

Similarly, the programmer might create subclasses for other types of books in the library. Note, however, that subclasses should not be confused with categories. Subclasses should be created only when they represent generic sets of objects that have properties and/or methods that are distinct from their parent class. Because subclass "reference" has a property (volume number) that is not defined for its parent class ("book"), it's a good candidate for a subclass. On the other hand, "fiction," "nonfiction," and "children's" are probably not good candidates for subclasses. They may represent different categories of books, but they have the same properties and methods as generic class "book."



A class' subclass and instances inherit the characteristics of the entire family tree.

In the same vein, a library application might include other classes, each with their own subclasses. For example, subclasses "VHS" and "DVD" might be set up for class "video," and subclasses "cassette" and "CD" created for class "audio." And classes "book," "video," and "audio" might themselves be subclasses under class "lending media."

By designing the application around a hierarchical, object-oriented model, the programmer makes it relatively easy to build the system progressively, in layers of classes, subclasses, subclasses of subclasses, and instances. Each layer inherits the characteristics of the entire family tree (all subclasses and classes) above it.

What Is OOP Not?

So, OOP is a magic solution, a panacea to all programming problems, right? Well, not exactly. OOP offers a new programming model that enables powerful new design and development techniques, but it does not provide solutions by its mere presence. As Andrea says, "An object-oriented application is only as elegant as its design." Just as there can be beautiful traditional systems, there can be ugly object-oriented systems. Neither traditional nor OOP programming models inherently provide elegance, but they both allow them.

Why OOP?

When the idea of OOP first started taking hold, the promise was for unprecedented levels of programmer productivity. Once the foundation was laid, the hype foretold, developers would be able to build new applications simply by picking from a mouth-watering smorgasbord of reusable components, a geek's buffet.

Has this promise been fulfilled? Umm, yes and no. Properly designed, an object-oriented system does yield productivity benefits for subsequent subsystems and enhancements. But, then again, so do properly designed traditional systems.

How about platform independence, another touted benefit of OOP? Again, yes and no. Java and C++ are both object-oriented languages. While Java offers true platform independence, experienced programmers say that up to 40% of a C++ application must be revised when porting it from one platform to another.

It may be that the best and most lasting benefits of OOP didn't appear on the early tout sheets, because they were realized only after years of experience. One of these benefits is wonderfully simple and decidedly nontechnical: the enhancing of plain old, garden-variety communications. Given that the OOP model more closely mimics the real world than does the traditional model, communications among techies and nontechies alike are facilitated, as OOP provides a common language. OOP as Esperanto.

Andrea presented an experience of a four-way dialog among programmers, a writer (herself), a customer service representative, and an account manager. The customer service rep viewed the world in terms of transactions, because that's what she did all day. The account manager saw things in terms of objects, such as accounts and contracts, because those were the planets that orbited around his world. The OOP programmer, of course, perceived both objects and transactions as programming tasks.

As a writer faced with the task of tying it all together, Andrea discovered that even a little OOP education went a long way. Not only did it provide a common language for all parties, but also, as Andrea says, "the OOP model is an effective way for you, as an information developer, to learn and to design your document." In other words, you can apply the lessons of OOP to your documents themselves, whether or not your documents describe OOP applications!

This brings us to the latest, and greatest, benefit offered by OOP—the ability to adapt. In today's business climate, the only constant is change. Whole companies, and the products and services that come with them, can be acquired and merged almost literally overnight. Companies comfortable in the widget business suddenly find themselves in the framistan product business and the follimating services business as well.

CEOs and Boards make changes that either used to take months or years to

Continues ☞

Comes to Terms - Part II

by Gabrielle de Serres

The second part in a two-part series on Wayne Gibson's presentation at the April STC meeting

Acronyms for Telecom Technology *(continued)*

CLASS - customer local area signaling service. CLASS includes features such as *69, automatic callback, and line identification known as caller ID.

Class of Service - the privileges and services each phone receives. For example, a restriction can be placed on calling outgoing numbers, so that employees cannot call 900 numbers from a business. Class of Service differentiates between business and residential users. Before phone lines were needed for data transmission and before public use of the internet, typically business phone lines were more available than residential lines. A hundred residential customers might have shared 10 circuits, whereas the same number of business customers would have shared 40 to 50 circuits.

Class of Office - the rank of a central office (CO), from 1 to 5, according to its size and function. Class 5, or end offices, are smallest, and provide dial tone out to the house. Several such COs exist in Sonoma County. Phone calls go from the house to a Class 5 office, then up the hierarchy as needed. Class 4 and Class 3 offices are called tandem offices. The one Class 4 office in Sonoma county is located in Santa Rosa. Our closest Class 1 office is located in San Francisco.

With the modernization and drastic size reduction of telephone equipment in the COs, phone companies have sold some of the older COs, or leased space out to other phone companies such as MCI, Sprint, Worldcomm, and AT&T. The other companies then have points of presence (POP) in the central offices in which their equipment resides.

DSL - digital subscriber line. DSL carries both data and voice on the same line without one interfering with the other. This allows customers to surf the net and talk on the phone at the same time using a single line. The data line remains connected (no need to dial in and wait for the handshake to establish a connection) and there is no impact on the switch in the CO.

Using current technology, DSL circuits can only be provided within a certain distance from the central office. In North America we use digital loop carriers (DLCs) and T-span circuits. Companies here in Telecom Valley (Sonoma County) are developing systems to put DSL services on remote access units.

Access - both as a noun and a verb, a word used abundantly in the telecom industry. *Newton's Telecom Dictionary* contains three pages of definitions, listing 44 separate items for this entry. Most entries, of course, have to do with gaining access to something. A common general usage in telecom is "access" refers to the equipment involved getting the phone call from the house or business to the CO.

BORSCHT - for most people, this is a tasty Eastern European stew. In telecom terminology it stands for battery (supplied to the subscriber line), overvoltage protection, ringing (as supplied to the line), supervision (for example, how does the phone know when to stop ringing), coder/decoder, hybrid (2-wire to 4-wire conversion), test.

Traffic Terms

Traffic refers to a flow of activity across a device such as a circuit, line, computer, or switch. Some traffic terms include:

(call) attempt - a change in the condition of the line, such as going off hook (taking the phone off the receiver and getting dial tone).

holding time - the time from when a receiver is picked up to when it's hung up. For phone companies, phone conversations (even between teenagers!!) aren't nearly as much of an issue as customers who surf the web for hours and hours.

busy hour - the busiest hour of the day, week, month, or year. Determining the busy hour is critical for traffic engineering.

busy day - the busiest days in the year, such as Christmas or Mother's Day. In traffic parlance phone switches function with a specified amount of traffic loss on busy days. This explains why we can sometimes get a busy tone even though the person we are trying to call is not on the phone.

ccs - centi call second (hundredths call seconds). Traffic machines monitor circuits and traffic operators analyze the results to determine the optimum equipment to handle the traffic.

Channel and Drop - terms used for carrier systems. A channel is a sub-unit of the main carrier system, and a drop is the access or delivered service that goes out to the house or business.

Fiber and Fiber Network Terms

Single Mode Fiber - a fiber optic line through which only a single mode of laser light can travel. The cross section of single mode optical fiber is about 9.2 to 9.5 microns in diameter. (A micron is a

Continues ☞

Writing for Programmers

Continued from page 4

accomplish, or more likely were not made at all. For example, an executive overseeing a mail order company with diminishing sales realizes that the company's future, if there is to be one, is as an Internet enterprise. Such a company cannot afford to wait three years for their software developers to redesign and reconstruct applications.

Although OOP is not a magic elixir for adaptation, it lends itself much more readily to a changing business model than does traditional programming. In this sense, OOP is not merely a nicety. It's a requirement.

(Continued in Part 3)

Come to Terms, Part 2

Continued from page 6

milllionth of an inch.) Single mode fiber is much more efficient than multimode fiber (see below) because it has less dispersion. Single mode fiber is now the standard for long-distance carriers. Single modes fibers have yellow casings. Currently, the high demand for single mode fiber means some companies must wait an entire year for orders of single mode fiber to be filled.

Multimode Fiber - a fiber optic line through which multiple modes of laser light can travel. The cross section of multimode fiber is 64 microns, considerably bigger than single mode fiber. Multimode is used exclusively for data, and is less expensive than single mode fiber. Multimode is not used in long distance because of the problem of light dispersion. Multimode fibers have orange casings.

Dark Fiber - fiber in the ground between buildings that is rented or bought from the phone company without any "light" service supplied on it by the phone company. Companies renting or buying dark fiber put their own services on it between their own buildings.

Other Fibers - Short fibers that serve different purposes include jumper fibers, pigtails, and stub cables. Stub cables are fiber cables with connectors on one side.

Working with fiber is more challenging than working with copper. Fiber splicing must be completely precise so that there is no loss of refraction of light as it passes through the fiber. The most common method of splicing fiber is fusion splicing, in which the fiber is cut with a diamond scoring tool, put under an arc, melted, and fused together. A mechanical device can also be used for fiber splicing; this device aligns the two pieces of fiber absolutely perfectly.

FTTC - fiber to the curb

FTTH - fiber to the home

FTTN - fiber to the node, network or neighborhood

FTTB or **FTTP** - fiber to the building or premise.

Other Important Terms

Supervision - detection of the person being called picking up the phone or hanging up. Supervision is used for billing purposes.

Edge Switch - termination switch in a broadband network.

Frame Relay - a Layer 2 protocol (see OSI model in part I in the May 2000 issue) for transfer of information between two compatible endpoints.

SMDS - switched multimegabit data service. A high-speed data transmission method, SMDS came into use in the early 1990s. It has not gained popularity, particularly with the advent of ATM and Frame Relay.

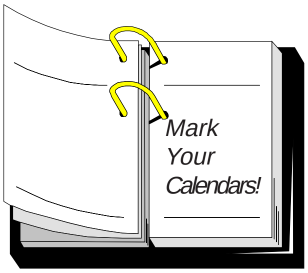
RJs (registered jacks) - telephone and data jacks registered with the FCC. About 39 RJ models exist, with the RJ-11 model being the most widely used telephone jack.

Epilogue

While I was putting together this article from the seminar that Wayne gave, I realized that not only is telecommunications terminology extensive and involved, but new terms constantly come into use. To further complicate the issue, older terms evolve and take on entirely new meanings. A person new to the industry has to both get a handle on the new terms and learn the evolution of terms.

To wade through the terminology, Wayne Gibson recommends that the first purchase of anyone working in telecom should be *Newton's Telecom Dictionary, The Official Dictionary of Telecommunications,* by Harry Newton, currently in its 16th edition. The author Harry Newton, like Wayne, has a great sense of humor. You won't be left falling asleep reading the *Newton's*. If only all dictionaries could be so entertaining.

Coming Soon!



October 19-21, 2000

STC Regions 7 and 8 Pan-Pacific Conference in Hawaii

For late-breaking information, see www.pan-pacific.org, and Jack Molisani's articles there.

Sr. Technical Writer Position

Job Qualifications: Proven technical writing experience. Experience with Microsoft Word, FrameMaker, and RoboHelp. Must possess strong project management skills and the ability to quickly learn new applications. Consulting experience a plus.

Job Description: Responsible for a wide variety of projects

Days/Hours: Full Time, Monday – Friday

Salary: Based on Experience

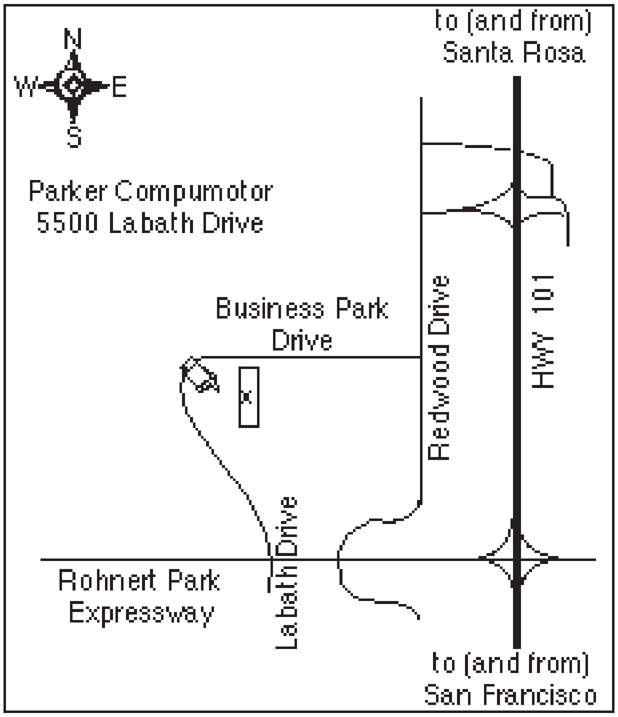
For information: Call Melanie in Human Resources at 707-838-0918.

We meet on the third
Thursday of each month

OurJuly Meeting

Thursday, July 20
To be announced.

Parker Compumotor
5500 Labath Drive
Rohnert Park
stc.org/region8/nbc/www/



northbay news
c/o
4389 sonoma highway
santa rosa, ca 95409