

Social Software: A New Generation of Tools

INSIDE

SOCIAL SOFTWARE

<i>The lessons of e-mail</i>	
<i>Can we get it right this time?</i>	
<i>Small is beautiful</i>	
<i>Box: Weblogs and wikis</i>	
Working With E-mail	9
<i>Kubi: Slipping in softly</i>	
<i>Shinkuro: Control your versions!</i>	
Working Outside E-mail	13
<i>Traction: Weblogs grow up</i>	
<i>Socialtext: Wiki, meet weblog</i>	
Platform Play	18
<i>CoSI: Cooperating systems</i>	
<i>Box: Small worlds and social networks</i>	
It's Not What You Know. . .	22
<i>Visible Path: Three degrees of connection</i>	
<i>Social-Software: Identity is two-way</i>	
<i>Box: The 800-pound gorillas</i>	
Take the Bad with the Good	27
Resources & Contact Information	29
Calendar of Technology Events	31

Help us keep PC Forum alive!
 Read and comment on Will
 Wright's speech on the Forum wiki:
www.socialtext.com/pcforum/
 Stay tuned for more transcripts. . .

BY CLAY SHIRKY

At Release 1.0, we've been writing about social software for decades, albeit under a variety of names. It comes in lots of guises, but the underlying principle has been that software should somehow strengthen and enhance human connections rather than impede them. That could not happen broadly until now, with the achievement of three necessary preconditions: First, people feel comfortable enough with technology to focus on the people at the other end rather than on the technology. Second, enough technology is now "standards-based" to enable new capabilities like social software to be adopted within (rather than apart from) a user's existing environment of software and data. And third, at least in the US business context, almost everyone a user could want to interact with is also on the Web - a fact that both produces the critical mass of users needed for social software to deliver and contributes to the ubiquity of standards.

Clay Shirky, as a longtime user and observer of the use of social software, excels in going beyond the above analysis of the necessary conditions for social software, to identify and promote what could be sufficient features to lead to its broad and productive adoption. These features are based on a better understanding of how people actually work in groups - or perhaps a modulation of the unrealistic expectations we had in the past. There was never quite the bubble for e-sociality as for e-commerce, but both fields have learned a lot by experimentation. (Fortunately, the education was not quite as expensive in the field of social software, which tends to attract social rather than commercial characters!)

In the issue below, Shirky outlines what he has learned, eloquently and lucidly explaining the tools and their uses. With appropriate discre-

{ continued on page 2 }

tion, he does not posit social software as a booming market, but rather as broadly available functionality that we will someday take for granted, just like ATMs or telephones (or the Web!).

– Esther Dyson

Release 1.0® (ISSN 1047-935X) is published monthly except for a combined July/August issue by EDventure Holdings Inc., 104 Fifth Avenue, New York, NY 10011-6987; 1 (212) 924-8800; fax, 1 (212) 924-0240; www.edventure.com. It covers the worlds of information technology and the Internet, including wireless communications, security, business models, online services, tracking systems, identity management and other unpredictable topics. . . and the policy issues they raise.

EDITOR-IN-CHIEF: Esther Dyson
(edyson@edventure.com)

PUBLISHER: Daphne Kis
(daphne@edventure.com)

MANAGING EDITOR: Christina Koukkos
(christina@edventure.com)

CONTRIBUTING WRITERS: John Hagel
(john@johnhagel.com),
JC Herz (jc@joysticknation.com),
Clay Shirky (clay@shirky.com),
Jeff Ubois (jeff@ubois.com)

CIRCULATION MANAGER: Natasha Felshman
(natasha@edventure.com)

SYSTEMS MANAGER: Beckie Jankiewicz
(beckie@edventure.com)

CONSULTING EDITOR: Bill Kutik
(bill@kutik.com)

Copyright © 2003, EDventure Holdings Inc. All rights reserved. No material in this publication may be reproduced without written permission; however, we gladly arrange for reprints, bulk orders or site licenses. Subscriptions cost \$795 per year in the US, Canada and Mexico; \$850 overseas.

We've known for decades that social software – software that supports group interaction – is one of the most profoundly important uses of the Internet. E-mail was the first killer app, in the 1970s. During the '90s, AOL established its dominance as an ISP largely because of its emphasis on social interaction – principally easy-to-use e-mail and chat. The recent study of US Internet use by Pew Internet and American Life Project notes that 84 percent of Internet users, or about 90 million Americans, say they have used the Internet to contact or get information from an online group, and 79 percent identify at least one particular group with which they stay in regular contact, making group participation one of the most common uses of the network.

Social software groups together several kinds of application, from online community applications to groupware to collaborative tools, but the common thread is that it amplifies or expands our social capabilities. As with anything social, there are good and bad aspects to this. (Flaming, the tendency of people to be more hostile in e-mail conversations than in real-world ones, is probably the best-known example.) Because it comprises all the complexities of group behavior, from collaboration to one-upmanship to backstabbing, designing social software is a problem that can't be attacked in the same way as designing a word processor. Designers of social software have more in common with economists or political scientists than they do with designers of single-user software, and operators of communal resources have more in common with politicians or landlords than with operators of ordinary websites.

The term “social software” describes patterns of use more than technologies, and has both consumer and business applications. Mailing list participants and spammers both use e-mail, but the spammers don't use it socially: They don't want to communicate with their recipients, nor do they want their recipients to communicate with

one another (lest a class-action suit results.) Social software includes everything from simple group e-mail to vast 3D game worlds like EverQuest. It can be as undirected as an AOL chat room or as task-oriented as an installation of Lotus Notes. Some types of social software are highly centralized, like WebCrossing's Web-based discussion forums, while others are decentralized and work to make the servers invisible to the users, as with Groove (SEE **RELEASE 1.0**, NOVEMBER 2000, MARCH 2001 AND JUNE 2001).

Social software is not a new concept. Both Douglas Engelbart and J.C.R. Licklider, early computing visionaries, talked about computers augmenting group interaction in the early 1960s. The surprise now is the renewed focus on this characteristic. The Web actually dampened the development of social software. Users kept using mailing lists and chat, of course, but most new software was designed for a one-way conversation between writers and readers of Web pages; two-way conversations were often an afterthought, with a BBS or "Contact us!" button tucked away on the side. Now, after years of sites and software designed to support big and largely disconnected groups, developers are working on social software again. This is in part because there are a number of interesting problems involved in helping people interact (identity, reputation management, conversational threading), and in part because the ubiquity of Web protocols means that developers can treat the Web as a platform.

Amplify and exploit

Getting social software right matters because even the largest organizations rely on small teams to do much of the hard work. Human networks have shown themselves to be redundant, resilient and flexible. Social software will be valuable to the degree that it can amplify and exploit these qualities. However, we haven't often gotten it right – yet. Businesses have typically invested in social software (néé groupware) that is aligned with management preferences for control over flexibility, often leading to software that is centralized, process-heavy and locked down: "If my software requires users to store all their files centrally, I'll never lose anything. If my software prevents users from sharing files with users outside my firewall, there won't be any leaks."

Meanwhile, the actual users of this software need to have group conversations without asking the IT department for help, and they need to converse, coordinate and share files with clients, vendors and partners outside the organization. Real-world collaborative patterns, in other words, are better supported by software that is decentralized, flexible and extensible.

In the same way the org chart never really describes the actual organization, rules for social software never describe its actual use. This is the story of social software: No matter what management wants to happen, the users get a vote, and whenever users are asked to adopt collaborative software that is difficult, complex or unfamiliar, they usually vote to keep using e-mail instead. Taking their cue from people's actual behaviors rather than some idealized projection, a number of startups are designing tools that help people get what they want from group interaction – sometimes interacting entirely online, sometimes bridging the gap between the virtual and the physical, and often supporting multiple patterns of use.

For example, weblogs (a lightweight publishing platform) can support both media (individual publications) and sprawling group conversations, and are spreading like wildfire. Wikis, a kind of collaborative workspace, are likewise spreading among distributed groups collaborating on projects from the creation of a free encyclopedia written from scratch (wikipedia.org) to a reference site for information architects (Iawiki.net). New real-time tools such as Hydra, a multi-person text editor, are starting to appear as group note-taking tools at conferences. The Emergent Democracy movement, founded by Joi Ito, hosts its meetings in a format called Happenings, where participants join a conference call and a chat room simultaneously, allowing them to carry on a two-track conversation in speech and text. The first Happenings involved two dozen or so people, several of whom have reported using the format for other conversations. It's not clear whether the exact mix of conference calls plus chat will catch on, but it is illustrative of the power now in users' hands that they can easily create such recombinant experiments. After several years during which the Web was used mostly in a "publish from the center/consume at the edges" pattern, there is an explosion of new software and new uses for many-to-many conversations.

It isn't clear that these new uses make for a good vendor's market for a variety of reasons, including a healthy skepticism on the part of clients, borne from claims made for previous generations of groupware. Then there is the possibility that established firms, especially Microsoft, will bundle social features into their platforms as users educated by the startups begin to want them. Nonetheless, a number of companies are betting that organizations need something that works better than "e-mail + attachments + IM" as the de facto collaborative suite and that they will be willing to pay for it.

The lessons of e-mail

If you want to understand social software, you must first understand e-mail. E-mail is the fundamental social application on the Internet and has a set of characteristics that make users prefer it for collaboration over many of the commercial alternatives, such as Lotus Notes. First, it works *with* the “personal” in “personal computer” rather than against it. Even though e-mail and IM are Internet applications, they feel local because they hide the complexity of the server infrastructure necessary for their operation, unlike applications that require the user to keep track of intranet addresses and logins. Second, e-mail maps well to social processes. As in the real world, a conversation is simply defined and initiated by its participants, rather than requiring the use of some pre-defined “space.” Third, e-mail carries data across organizational and technological borders; users invariably need to move information across firewalls and to have conversations with people other than their fellow employees, simply to get their jobs done.

These three effects – local, social, global – have created a virtuous circle where the spread of e-mail (and, more recently, IM) makes those tools both ubiquitous and familiar, further deepening their value and their hold. Furthermore, because they are so well understood, they are one of very few applications that users feel completely comfortable setting up and using on their own, making them the path of least resistance as a collaborative tool.

This explains the ubiquity of e-mail and IM: They are not the best tools possible, but they are the least-bad ones available. They have several flaws, of course – particularly e-mail, where the very ease of sending something off for modifications produces problems of version control. Worse, the older copies proliferate on recipients’ PCs, making it possible for several users to think they are looking at “the” document, while actually looking at subtly different versions. E-mail also makes security breaches easier, sometimes because it lowers the barriers to malicious use, but often because ease-of-use means ease-of-mistake (as with the famous Agency.com story in which the entire company’s salary list was forwarded to all employees). Finally, when most knowledge is in e-mail, it is not easily searchable, and the desire to delete in order to reduce clutter lowers e-mail’s value as a repository of project history. (SEE CATAPHORA IN **RELEASE 1.0**, MARCH 2003.) Despite these sorts of flaws, however, e-mail and IM are still at the core of most users’ work patterns, because none of the alternatives have the same flexibility and lightness. Until now.

WEBLOGS AND WIKIS

Though weblogs and wikis are not the only kind of social software currently in fashion, they are some of the most commonly used (and studied) ones, and they each demonstrate the value of putting dead-simple and Web-native tools at the user's disposal.

Weblogs (sometimes called blogs, from user experience guru Peter Merholtz's pun on "weblog" as "we blog") are everywhere these days. The simple pattern of weblogging - daily posts displayed in reverse chronological order - has turned out to be a terrifically flexible and broad way of handling all sorts of content. Though weblogs lack about 80 percent of the functionality of any self-respecting content management system, like many other classes of software (instant messaging, the Web browser, e-mail), weblogs work *because* they are so simple: They finally make the Web a writeable medium as well as a readable one, and they put publishing capabilities in the hands of individual users. The 20 percent functionality they provide provides 80 percent of the value of any publishing tool.

Some weblogs are informal conversations among small groups of friends. Some weblogs are broadcasts to the world, and mix personal and political observations. Others are group-run and topic-specific. And a few weblogs have become serious media outlets, with monthly traffic that rivals many big-city daily newspapers.

Most weblogging is pure publication, of course, with little social component. However, weblogs do have three critical social functions: First, they enable groups of users to bridge public and private conversations. LiveJournal, for example, allows users to create community weblogs where members are both conversing with one another and making the results of their conversations available to a wider public. (There is even a LiveJournal business community where LiveJournal business issues are raised and discussed in public.)

Next, most weblogging tools (Movable Type and Blogger seem to be the most popular) now provide a "comments" function, allowing webloggers to host conversations on their website. Readers can thus converse (or argue) with the original weblogger and with one another, creating a kind of fused publication + BBS. This doesn't work as well for, say, MSNBC, because its user base is too large and diffuse, a problem best explained by the "Small Worlds" network model. (See page 21 for more on the power of small group dynamics.)

blog2blog

Finally, there are looser conversations *between* weblogs. This is the most important social contribution of weblogs, as it bridges the gap between pure publication and tight conversation of the sort that takes place on mailing lists or BBSes. In the loose conversational model, one post to a weblog might be responded to on another weblog, generating further responses by both the original weblogger, and by new webloggers jumping into the fray. (The most organized example of this was N.Z. Bear's cross-blog debate earlier this year, where anti- and pro-war bloggers asked one another a set list of questions, and anyone who wanted to participate answered the questions on his or her own blog.)

While the ability to support so many kinds of communications makes weblogs the most widely adopted new piece of social software, they are not optimized for any particular collaborative pattern. Wikis, by contrast, are more useful for capturing a group's state of mind over time. Wikis were invented by Ward Cunningham, a programmer who works on large-scale distributed systems, who called the original version the WikiWikiWeb. (The term "wiki" means "quick" in Hawaiian.)

Like weblogs, wikis succeed by excluding almost every possible feature; in fact, there are just two critical wiki functions: editing an existing page and creating a new one. Any page posted is editable by any user: At the bottom of every page is a link that says "edit this page." Clicking the link opens a form with the text of the page included in it, and the user can add, alter or delete text. The new text then replaces the old.

It is easy to create new pages as well, simply by naming them. Most Wikis accept words in "CamelCase," as Cunningham calls internal capitals. When a phrase appears in CamelCase, such as SocialSoftware, the wiki will create a new blank page, called SocialSoftware. This "forward linking" turns a decade of website development practice on its head: There's no "website design." Users simply create their own pages as needed.

While all this freedom offered by wikis would seem to be a recipe for disaster, wikis that are frequently edited by their users are surprisingly stable and easy to use. Though the "every page is editable" principle would seem to expose every wiki to merciless drive-by damage, the software makes it easy to roll back to any previous version. Furthermore, would-be graffiti artists have no

WEBLOGS AND WIKIS (CONT.)

way of preventing their work from being "edited" in turn, making a wiki a particularly bad place to act out (unlike mailing lists, where users can respond but can't erase, leading to interminable flame wars).

Likewise, the ease of creating a new page would seem to make wikis an unnavigable mess, and in fact, most wikis do start out with confusing navigation early on. However, the ease of editing means that putting the site in order after the pages are built becomes quite easy; this process is often called wiki gardening. Furthermore, in addition to the link structure, the wiki sports a "recent changes" button, which displays a list of all the pages on the site, with the most recently edited ones displayed first. (This is more evidence of the importance of time in social ordering.)

Most "process" in businesses is put in place to

prevent users from doing stupid things, but also adds complexity. By being easy to use and making it easy to repair any damage, wikis get away with having almost no built-in process and are far easier to adopt and adapt than "serious" tools such as Lotus Notes or even Groove.

A problem both wikis and weblogs face in the enterprise is "attention management." One of the signal virtues of e-mail is that people are always scanning it. People will have to learn to check wikis and weblogs as well. As Irene Greif, an IBM Fellow who heads the Collaborative User Experience Group, says, "Team spaces have often failed because they tend to make information be out of sight and out of mind. So, as exciting as wikis and weblogs are to dedicated users, one question will be whether they can grab enough attention on an ongoing basis to become an important force in corporate networking."

Can we get it right this time?

Is there any reason to think that this generation of tools will avoid the fate of its groupware predecessors? There are three kinds of advantages that this generation of tools enjoys: familiarity, standards-based Web technology and broad availability to anyone who can get onto the Web.

Familiarity is the easiest advantage to see. Every piece of technology needed for weblogs (SEE BOX, PAGE 6) existed by 1994, when the first browser that supported forms came out. The pattern of weblogging appeared a couple of years later, most notably with Matt Drudge. The first formal weblogging platform appeared in 1998, with the development of Blogger (recently acquired by Google), but it wasn't until 2002 that the general public began to be aware of them. Weblogs took eight years to go from technological possibility to widespread use (the four main weblogging platforms – Movable Type, Blogger, Radio Userland, and LiveJournal – now have over a million active accounts) because social patterns change slowly, even though software changes quickly. We've now had the basic interfaces for social tools around long enough that many people are willing to adopt them.

This generation of tools also has several technological advantages over earlier versions of groupware and other social software. Most importantly, they are Web-native: They generally take the browser interface as a given, as do most users. Previous Web-accessible versions of groupware were often giant installations designed to work inside the corporate firewall and later retrofitted with a Web front-end, a combination that rarely achieved the ease of use of, say, weblogs designed

around the Web from the start. In addition, newer tools almost always use http as a way of providing cross-platform, firewall-indifferent communications. Though not many people building social software speak in terms of “Web services,” many of them provide service-style interfaces: programmable, Web-accessible interfaces that return data structured in XML.

As a consequence of being Web-native and service-oriented, this generation of social software fits the “small pieces, loosely joined” pattern so beautifully articulated by David Weinberger in his book of that name. Rather than attempt to provide all functions to all people, the tools and services being developed can be combined easily and as needed, without having to be formally merged. The combinations can happen between packages, by tying things together either in one interface (a Web page that points to multiple resources) or by connecting the output of one program to the input of another.

This Web plug-and-play pattern is best demonstrated by the astonishing success of RSS (Rich Site Summary or Really Simple Syndication, depending on who says so). RSS was invented by Dave Winer and Userland software (SEE **RELEASE 1.0**, JULY 1999), and makes the content of a web page or other document available in XML. Though it was initially designed to syndicate published documents, it has been pressed into service by the weblog world as a way of syndicating conversations as well.

Small is beautiful

There is another commonality among the current generation of companies building social software, a commonality that has more to do with design philosophy than technology: Most of these companies assume there is an inverse ratio of value to scale. This is different from the story of the Web, which was the story of explosive growth: 30 million Web users! 100 million! 500 million!

The growth story became the story of individual sites as well, as the most common metric for success became “How many?” How many readers did MSNBC have? How many users did Yahoo! have? How many customers did Amazon have? As exciting as this up-up-up period was, it ignored a basic human pattern: Too many cooks spoil the stew. If you want to get something done, you give it to a small team, not a whole department. If you want to kill a meeting, invite a couple dozen people.

In human connectedness, less is more. The primatologist Roland Dunbar asserts that the human mind is optimized to keep track of groups of 150 or so members, an

idea popularized as “The Rule of 150” in Malcolm Gladwell’s *The Tipping Point*. You can see a similar pattern if you examine your Rolodex: Your 1000 contacts, your 150 friends, a dozen or so close friends, the two or three people you would donate a kidney to. The more valuable the association, the smaller the number of people in that group. By accepting that small, focused groups have different kinds of behavior and different needs and produce different kinds of value from large groups, businesses such as Traction (weblogs designed to be used by “groups with a goal”) or Shinkuro (support for small group document sharing) are able to offer users services they can’t deliver through one-size-fits-all websites.

The companies below are all betting that a focus on the peculiar needs of group interaction and on simplicity in use will enable them to succeed where others have failed. They represent a fairly wide spectrum of approaches, though the goal is the same. Sometimes this support is direct (Kubi, Shinkuro, Traction, Socialtext), sometimes it involves the creation of a platform (CoSI), and sometimes it means using Social Network Analysis to create value (Visible Path, Social Software).

Software that supports group interaction covers many more kinds of companies than those listed here. There are dating sites such as Match.com and Spring Street Networks, massively multiplayer game environments such as EverQuest and Star Wars Galaxies (SEE **RELEASE 1.0**, OCTOBER 2002), and companies using software to improve real-world social situations, such as Meetup and nTag (SEE **RELEASE 1.0**, MARCH 2003). The companies listed below all offer social software that creates value in a business context.

Support for Group Interaction: Working with E-mail

All enterprises have more knowledge in their employees as a group than any one person, even (especially?) the CEO. The worst case is where one person has a problem and another knows a solution, but neither knows the other – or that the other knows. Despite e-mail’s advantages for communication, it falls down as a close collaboration tool on complex projects: E-mail makes it hard to keep everything related to a particular project in one place; e-mailed attachments can lead to version-control nightmares; and it’s almost impossible to get the Cc: line right. If the Cc: line is too broad, it creates “occupational spam” – messages from co-workers that don’t matter to everyone addressed. If the Cc: line is too narrow, the activity becomes opaque to management or partners.

In the past, we endured these problems because e-mail's virtues far outweighed its vices. The two companies in this section offer solutions not by building all-singing, all-dancing groupware suites, but by offering lightweight solutions that complement e-mail. Kubi embeds better collaborative tools in e-mail itself, while Shinkuro plans to make e-mail the transport mechanism for a new collaborative platform.

Kubi: Slipping in softly

Kubi Software, based in Lincoln, MA, understands the critical importance of e-mail and has launched a collaborative application, Kubi Client, which works with Microsoft Outlook and Lotus Notes. "We solve an eternal collaboration problem – making teams of people work more efficiently and effectively," says Julio Estrada, Kubi's founder and CEO. "But instead of forcing individuals to leave e-mail, we bring context and structure to that familiar environment."

The Kubi Client, launched in late April, creates shared areas, called "Kubi Spaces," containing a variety of information types including discussion threads, documents, tasks, calendar events, time lines and contacts. Any resemblance to Groove Spaces is not coincidental, though Kubi's "Spaces" differ in that they comprise a set of func-

tion-specific e-mail folders. Upon installation, the software represents itself as a folder in the e-mail hierarchy. The project leader decides what kinds of data others may access and who's invited. The use of the e-mail folder as the logical site for storage, instead of updating individual e-mails, saves the user from having to continually check the same message to see if anything has been added, a problem with Zaplet (SEE **RELEASE 1.0**, JUNE 2001). (DISCLOSURE: EDVENTURE HOLDINGS IS AN INVESTOR IN ZAPLET.) By opening the e-mail folder containing a Kubi Space, a user should be able to participate in or track what is going on with a particular project.

Everything a user posts into a particular Space is automatically replicated via SMTP (simple mail transfer protocol) to the equivalent spaces of the other participants. (This pattern of store-and-forward update is also used by Groove and Shinkuro, profiled below.) Individuals decide what specific data they want to share when it comes to such things as calendar events and personal contacts. Because it distributes data over e-mail, which traverses firewalls, Kubi automatically encrypts all data for transmission, using public-key cryptography. The sweet spot for the number of participants in a Kubi

KUBI SOFTWARE INFO
Headquarters: Lincoln, MA
Founded: July 2001
Employees: 35
Funding: \$8 million from Lazard Technology Partners, Venture Investment Management Company and individuals
Key metric: 300 organizations partici- pated in pre-release trial
URL: http://www.kubisoftware.com/

Space, Estrada says, is from two to about 20 people, which echoes the experience on mailing lists and other conversational channels such as irc.

Estrada, formerly chief architect and development manager of Lotus' QuickPlace Web collaboration product and, before that, lead architect of Lotus' Domino Web server, cites his new company's communications with patent lawyers as an example of a useful collaboration with the product. He says, "A number of architects in the company are working with intellectual property attorneys in Washington in a Kubi Space where we post design documents and discuss strategies around each patent. Everything is secure. We're not concerned about someone intercepting it."

About 300 companies tried the software during the beta period, and Kubi is now trying to convert them to paying customers. The decentralized architecture, which helps make the product both cheap to run and tolerant of partly-on nodes such as laptops, made some potential clients nervous even with its automatic encryption. "Enterprises are not interested in a solution unless it's a solution where they control the data," Estrada says. To address this concern, Kubi is building a server edition, expected to ship in July.

The current version is Windows-only and requires that the user have Lotus Notes or Microsoft Outlook installed. It costs \$149 for a single user, with discounts as numbers of seats rise. A 10,000-seat installation would come in at about \$40 per seat, Estrada says. The company is making a 30-day trial version available, meaning existing users can invite new users to join Kubi spaces with them. When one user invites someone into a shared space, the other person doesn't have to pay for the client software; only if that second person creates his own space to use with other people does he or she then have to buy a license.

Shinkuro: Control your versions!

Shinkuro is part product and part platform, having positioned itself at the place where the user interface and the underlying mechanisms of the network meet. The essence of collaboration is sharing, and in most organizations what gets shared is files: written documents, presentations, spreadsheets. Founder and CEO Steve Crocker, one of the early creators of the Internet, says, "The Internet was created to help people work together, but it's still surprisingly hard to engage in cooperative work over the net. E-mail with attachments is the de facto mode of cooperative development of documents. Compared to what's possible, this is crude, inefficient and far less useful than we could – and should – have."

File sharing and version control is the axle on which collaboration turns, with the file in question serving both as the repository of the work and the package that gets transported and transformed, often by several people in parallel: “I wrote a rough draft, sent it to the boss for comments, to legal for some boilerplate, and to the comptroller for the real numbers.” As noted above, the very ease of e-mailing a document around can create version control problems.

Shinkuro is approaching this problem with a simple premise – e-mail is at once the problem and the solution. From Crocker’s point of view, e-mail is the right mechanism for the transport of files, but the wrong mechanism for managing those files once they arrive. Shinkuro’s approach is to set up a folder where a controlled group

of users can share files. This folder is where files handled by Shinkuro are stored and managed. When a user creates a new Shinkuro group, she can invite other users to join her. (Users who aren’t using Shinkuro yet are invited to download and install the software first.) Files put into a Shinkuro folder by any user in that group are sent via e-mail to the other group members. These e-mails are then intercepted on the recipients’ computers and the new version of the file is simply filtered into the right folder. The user can be alerted to new or updated files in several ways, including flags on the files, the Shinkuro application dashboard and an icon in the Windows system tray. (The software is Windows-only, but Mac and Linux support is expected in the next release.) “Keeping the user well-informed is a key part of the Shinkuro philosophy,” says Crocker. By putting the files in the file system but making alerts to

changes visible in several places, Shinkuro saves the user the hunt through hundreds or thousands of mails searching for a particular attachment – and ensures that the user gets (only) the latest version.

Because Shinkuro uses e-mail as the transport mechanism, users don’t have to be online at the same time. It also solves some problems inherent in sharing via e-mail: Faced with an 11-megabyte PowerPoint file, Shinkuro slices it into 11 one-meg files, and reassembles them into a single document on delivery. This enables Shinkuro users to share large files even with a finicky e-mail gateway (or a restrictive IT policy on attachment size). This is like a higher-order version of the subdivision of data into packets that makes the Internet work in the first place.

Finally, Shinkuro makes sharing files by e-mail more secure than today. “Safety and security are paramount,” says Crocker. “Files are never lost or damaged and all

SHINKURO INFO
Headquarters: Bethesda, MD
Founded: February 2002
Employees: 5
Funding: undisclosed amount from private sources
Key metric: DARPA funded Shinkuro for research on “collaboration between individuals in different networks and geographic locations”
URL: www.shinkuro.com

transmissions are encrypted until they get to the recipient's machine. Users continue to work in ways to which they are accustomed, and we take care of the details in the background."

As with Kubi, comparisons with Groove are obvious. Like Groove, Shinkuro provides secure, serverless work spaces where groups can share files. Though the usage pattern is the same, Shinkuro differs from Groove in several respects. Most importantly, it leaves messages in the e-mail box and files in the file system, rather than asking users to adopt a new interface for managing a separate space. Also, Shinkuro considers the existence of separate versions of a file a feature, rather than a bug. Where Groove goes to great lengths to synchronize group edits, Shinkuro saves different versions of an edited file in sub-folders labeled by group member rather than trying to reconcile conflicting versions automatically, because knowing who edited what is valuable information that the user may prefer to process manually. Finally, it uses the existing e-mail system for its store-and-forward capabilities, saving it from having to build and maintain Shinkuro-specific servers.

The system is still in development. Early work on the system was sponsored by DARPA's Advanced Technology Office, and the software is being improved based on feedback from alpha users. (As a good sign of viral spread, Crocker says Shinkuro can no longer track all the alpha users, as the original test group sent copies to friends and colleagues.) It is planning a beta launch in the second half of this year.

High on Shinkuro's to-do list is support for additional modes of transport. Though e-mail is the obvious first target, the company wants any two Shinkuro users to be able to share documents using ftp or http as well. Because of its focus on offering a platform rather than an application, Shinkuro is also working on ways to expose its functionality to third-party developers.

Support for Group Interaction: Working Outside E-mail

The next two companies are also building lightweight solutions, but instead of locating better collaboration tools in e-mail, they are seeking to make tools that co-exist with e-mail. Traction provides enterprise weblogging as a way to solve the Cc: line/annotation and sharing problem, and Socialtext is building a collaborative platform on top of a wiki.

Traction: Weblogs grow up

Traction Software, a startup in Providence, RI (near the Brown University campus, where many of its founders studied), wants to bring the weblog revolution to the enterprise. Weblogs hold some obvious attractions for a business setting. Traction (the name of both the company and the tool) organizes information by group and time, makes material widely available, and invites employees to share information. As Greg Lloyd, Traction's founder and CEO, puts it, "We want to make it easy for individuals to create or comment on content in both public and private spaces. We also want to let them see the union of all conversations and activity in those spaces, organized by project, importance and time." These features hold the largest potential for weblogs to transform business environments, because they function as what Cory Doctorow calls his "outboard brain," providing a way for groups to pool individual knowledge. (Doctorow, Outreach Coordinator at the Electronic Frontier Foundation, also runs the influential weblog BoingBoing.net.) Lloyd sees this aggregation of dispersed intelligence as a core part of Traction's value: "By championing the personal voices within an organization, and by helping users cite each other's writing, we can help create streams of thought and opinion that can be aggregated from the bottom up."

Traction attempts to utilize the simplicity and ease of use of blogging software to publish and annotate content, while allowing better control of security and access than standard blogging tools. The tool lets employees report their efforts and observations in a weblog. Other interested parties (and, in the case of sensitive material, only those parties) can then have access to that material in one place and on-demand, rather than piecing things together from a dozen Cc:ed e-mails.

The front page of a Traction blog is the same as any weblog: recent posts of relevant material, listed by group, in reverse chronological order. These posts can be links to external material ("Check out this interesting *Forbes* article on one of our clients"), internal material ("Here's our current marketing deck. Comments?") or pure commentary ("I posted my thoughts on our current product mix"). Traction allows the creation of a far more complex mix of posts than the average weblog tool, however, by cross-referencing posts by groups and user-defined labels. Every user of the system is a member of one or more groups: A user could be a member of the Sales group, the Directors' group, and the Chicago office group. Both individuals and groups have a set of permissions relative to other individuals and groups. Someone in the Sales group might be able to post in the Sales weblog, to append comments or questions to existing posts in the Product group's weblog, and to read the CFO's weblog, but might not have permission even to read the Board group's weblog.

(Users can ask to be notified of updates by e-mail so they don't have to be in constant scanning mode.)

Traction also provides a much richer set of annotation features than most weblogs. Each item posted to a Traction blog can be flagged with any number of user-created categories, such as Urgent, or relevant to a particular group, such as Executives. (And, true to form, Traction allows you to set who can create new markup categories for which sorts of posts.) In addition, posts can be saved, copied, or forwarded by e-mail with or without accompanying annotations, allowing a user to send around a copy of potential sales material while hiding the VP's scathing commentary. And, as icing on the cake, all the various markup, annotation and other manipulations treat the paragraph as the logical unit of the system, allowing for granular commentary where needed.

Lloyd says, "Everyone asks how we can manage a mix of material sorted by importance, area of focus and time, as if this is an unsolvable problem. But newspapers solve it every day, by accepting that there is no one answer, and by making up a new front page every day. In groups, coherence comes from shared labels: We all agree how something should be characterized, so the solution doesn't need to be perfect to be useful. Given the volume of material in the average newspaper, which readers are perfectly capable of navigating through, we think this is a solvable problem."

One big challenge for an enterprise weblog is getting users to use it: Most public weblogs languish after a few posts (much as most personal diaries peter out after a few entries). Paul Perry, an IT director at Verizon, a Traction client, says that getting employees to use it has been relatively simple: "People see that if they want influence in the company, all they have to do is post." The trick will be not so much weaning employees off e-mail (an impossibility, for good reason), but rather convincing them to use e-mail to notify them of new content on the weblog.

Traction overcomes the risk that if employees feel they are being forced to say everything out in the open, they may say nothing at all (or they'll restrict their comments to the "Ooooh, Ms. CEO, you are so smart! Can I please have a raise?" variety). Traction's answer to this problem is to allow groups some freedom in creating Traction spaces: "Each project space has its own team and its own audience. Project spaces can be opened up to a wider group of peers or senior management, or kept private, as the project group wants." Lloyd uses Traction's own customer relations

TRACTION SOFTWARE INFO
Headquarters: Providence, RI
Founded: November 1996
Employees: 9
Funding: \$1.7 million from In-Q-Tel, Slater Center for Interactive Technologies and individuals
Key Metric: 25 enterprise and govern- ment customers
URL: http://www.tractionsoftware.com

operations as an example: “We create one project area for each customer, visible only to that customer and us, as well as one group project for all the customers. When we have a new announcement – an updated SDK, say – we post it to the group project and everyone sees it. But if one of those customers wants to talk to us about helping them implement it, that conversation happens in our separate shared project, away from the other customers.”

Traction avoids the trap of the monolithic app by providing a variety of skins to ensure that it can interoperate with legacy applications. Although a “skin” usually refers to a particular look and feel for an application, Traction’s skins can include behaviors (for example, the concept of a specific fiscal year reflected in the way quarters are selected and summed) as well as pointers to other applications. Verizon has a skin that imports many other Verizon-specific intranet functions, such as search and access to the employee database.

As with almost all social software, Traction faces hard choices between user enthusiasm and purchaser buy-in. The simplest way for an application to spread is to be freely available (c.f. ICQ and Napster). The simplest way to keep an application from spreading is to make it cost too much, in either money or time. Traction is trying to split the difference by using the asymmetry of the publishing pattern: few writers, many readers. If you want to follow the conversations hosted on Traction, there’s no charge, but if you want to participate with your own comments, there’s a fee. Traction is sold as enterprise software, either run as an ASP or hosted in-house, with the usual installation and per-seat charges for the number of users who can create, import or annotate content. This split between reader and user makes it simple and cost-free for a firm to expose Traction pages to the world (or even just to another organization) as a way of sharing collective intelligence.

Socialtext: Wiki, meet weblog

Socialtext was founded with the goal of helping organizations take advantage of “simple tools that people [would] actually use,” says Ross Mayfield, co-founder and CEO. (DISCLOSURE: CLAY SHIRKY IS ON THE ADVISORY BOARD OF SOCIALTEXT.) Taking a cue from users’ rejection of most groupware solutions, Socialtext is creating tools for collaboration in the enterprise by identifying what software works today and combining and improving upon those solutions rather than re-inventing them. The technology at the core of Socialtext’s offering is a wiki. The company also uses the expression “collaborative workspace” to refer to its software. (What’s lost in poetry is gained in clarity.) The developers of Socialtext started by identifying useful features

from several existing wiki platforms. They then created their own version of a wiki, adding additional features attractive to enterprise users, such as page-level user categorizations (as in “File this page under Products, Competitive Research and John’s pages”), the ability to both post material and receive notification of recent changes via e-mail, and support for RSS feeds.

Mayfield sees Socialtext as complementing portals and other enterprise software, not replacing them. “The value in portals comes from people who are paid to organize information and systems taking lots of structured data and presenting it in certain formats,” he says. “We get people to participate who wouldn’t otherwise. We capture unstructured or semi-structured information, and give them incentives to share it.”

Socialtext has focused on integrating wikis and weblogs, even though the patterns supported by the two kinds of software are quite different. The two key attributes of material in a weblog are Who said it? and When?, while these labels are either less important or missing altogether in a wiki. While there is a presumed fixity once something is posted to a weblog, the editability of a page is key to a wiki. Despite (or because) of these differences, Socialtext believes that by treating the wiki and the weblog as alternate views of the same material, it can provide users with a single package that supports both collaborative and published views of the same material.

One classic charge to a group within an organization, whether formal or ad hoc, is, “You guys work on this problem, and when you’ve figured something out, come tell us.” Everything from product design to pitch documents follows this pattern: Assemble a group, have a conversation, publish a document, gather comments, repeat.

Wikis are good for groups that want to brainstorm or to converge on some shared sense of a particular problem, but they are poorly suited for publishing the results in a fixed form or soliciting comments from outsiders. Weblogs are the opposite: poor for shared effort, great for publication and solicitation of comment.

Socialtext aims to fuse the two so that a group can come together in a wiki, organize its thoughts, agree on some formal way of presenting those thoughts, and then expose that page as a weblog entry for comment by the rest of the organization or the rest of the world. “We’ve blurred the difference between a wiki page and a weblog post,” Mayfield says. “The weblog is just a reverse chronological sorting of wiki pages within a given category, whereas the wiki is a logical sort.” True to its vision of doing the simplest thing that could possibly work, Socialtext makes creating a weblog view

SOCIALTEXT INFO
Headquarters: Palo Alto, CA
Founded: December 2002
Employees: 4
Funding: undisclosed amount from angels
Key Metric: 20 operational deployments
URL: http://www.socialtext.com

of a wiki page as simple as placing it in a category that has the word “blog” in it: The software takes this as a command to create a new weblog and publish that page.

Conferences and other real-world gatherings, especially ones with WiFi connectivity, allow conference organizers to offer wikis as a kind of shared scratch pad for attendees to share their bios, pointers to interesting material, and thoughts on the conference. In fact, Socialtext created a conference wiki for PC Forum (SEE RESOURCES). Forum attendees posted materials relevant to the conference itself, and then took the wiki into “virtual corkboard” territory, arranging evening events and rides to the airport. One long-time PC Forum devotee who couldn’t be at this year’s conference (Keith Teare) even logged in remotely and created his own page, carrying his greetings to the conference attendees. Likewise, scenario-planning firm GBN has begun using a Socialtext wiki in some of its client meetings, to capture the proceedings in a central location.

The Socialtext user interface is still fairly raw, Mayfield acknowledges. That reflects, in part, the startup’s small size and bootstrap funding. There’s no built-in search functionality, either. Mayfield expects customers to integrate Socialtext with the search tools they’ve already installed on their intranets.

Socialtext sells both product and services. The biggest client, a major hardware and software vendor, is paying for software and service and building a giant Socialtext site to support its developer community. The product revenue model is fairly typical: per-seat pricing of \$30 per month per user, with volume discounts. Socialtext expects many of its installations to grow organically. To aid that process, it is also offering a “starter kit” version – a five-user workspace, licensed for a year for \$995. The service model is more complex: It includes not just software use but also advice about how to take advantage of social capital within an organization using Socialtext.

Platform Play

The Web has both proven the value of the “small pieces, loosely joined” design model and made it easier to adopt, by providing http as a ubiquitous transport mechanism. This in turn makes platform plays easier to build (though not necessarily to charge for), because anything that relies on http finds much of the infrastructure already in place. As people building social software abandon the idea of one-size-fits-all solutions, providing a platform for others to build on or hook into

becomes an obvious alternative. Users are generally attached to their existing tools (and perhaps most importantly, their existing interfaces). Creating software that operates “under the hood” offers a way to introduce new capabilities without requiring the users to alter their daily habits. CoSI provides a platform that allows clients to integrate social functions into their existing infrastructure.

CoSI: Cooperating systems

The personal-computer revolution, at its beginning, was about what individuals could do with the tools they had on their desktops. “Now [PCs are linked and] these tools are becoming networked applications,” says Kai Gradert, co-founder, President and CTO at Cooperating Systems, Inc. (CoSI). While it’s natural for developers to bring communications to their applications, there’s no need to re-invent basic infrastructure for every application. Instead, CoSI, based in Santa Barbara, CA, is building a software layer, above the operating system but below the applications, that allows developers to embed basic social communications in any application. “We’re building the first post-browser platform for rich personal communications,” says Gradert.

He and co-founder Phil Clevenger spent years with MetaCreations, the graphics toolmaker. “The communities that sprang up around those tools were highly motivated and knowledgeable,” Gradert says. MetaCreations ended up incorporating communications features into its desktop software – “stand-alone desktop applications taking advantage of the Web infrastructure” – but found that it could not rely on the browser because it lacked the functionality they needed. Having lived through the pain of inventing community functions once, they saw an opportunity to spare other developers that same agony.

HelloWorld is CoSI’s flagship product, now in preview release to a developer community experimenting with it. It comes at the issue from the opposite direction that the MetaCreations teams faced. Gradert says developers shouldn’t have to worry about creating a communications infrastructure before designing an application; it should be waiting for them before they begin. HelloWorld, he says, is that infrastructure. You can browse for hours in what Gradert calls the Web’s “cold information space.” There’s nothing wrong with that, but CoSI’s team says the “next wave is rich personal applications – gaming, webcamming, narrowcasting, photo-sharing – all done on the desktop layer.” It’s people space, not information space.

COOPERATING SYSTEMS INC INFO
Headquarters: Santa Barbara, CA
Founded: September, 2001
Employees: 7
Funding: \$850,000 from Santa Barbara Technology Group and angels
Key metric: currently seeking \$3 million in Series A round
URL: http://www.cooperatingsystems.com

HelloWorld comes with some modules for basic functions other than communication, such as messaging and data transfer, though CoSI expects much of the functionality to be written by users and third-party developers. The core of HelloWorld's platform is the HelloWorld Open Protocol Exchange, or HOPE. (The company's terminology abounds with this sort of wink. Its URL, cooperatingsystems.com, can be read two ways; HelloWorld is a reference to the canonical first program in any language; and CoSI is pronounced "cosy.")

HOPE provides the kind of synchronization services that make IM buddy lists run, though it can coordinate much more than chat. Built-in tools so far include chat, photo- and document-sharing and Web serving. Users download and install a desktop application. (At this stage, it's Windows-only, but a Mac client is coming. CoSI also wants to build a Linux release, depending on available resources.) Users and groups can be added easily to ad-hoc networks. The software supports a variety of network connection models, taking firewalls into account. Perhaps most importantly, it does not require a centralized (read: expensive) database to keep track of users.

Visual cues are everywhere in the system. For example, CoSI's "geo-contextual user interface" (a play on GUI) is a set of maps and mapping tools that let users communicate in cyberspace while keeping track of one another in geographical space. The user sees maps of both the technological and social aspects of the system, from the location of network nodes and operations, to the location of correspondents or the origin of messages. Images are used throughout to express sentiments and ideas. HelloWorld users show up on-screen as icons, and users can configure multiple avatars to represent themselves in the HelloWorld identity system. Whenever a user updates his identifying image, it is immediately updated throughout the system, allowing the user to signal not just identity but also mood or status, such as "Don't bother me" or "Away from my desk."

CoSI, which has been working on the product for several years, is planning to assist developers with a series of application programming interfaces and toolkits. Right now, the user base consists of classic early adopters, whose enthusiasms are evident on the company's discussion boards.

SMALL WORLDS AND SOCIAL NETWORKS

Accompanying this generation of social software is new research into the ways groups work. As hard as it may be to believe, we had no good way of modeling large-scale social networks until 1998. Prior to that year, the classic model for social network theory was something called a Random Network, developed by Paul Erdos in the early part of the 20th century. (The name alone should tell you something about its inapplicability to human networks, which are distinctly non-random.) We've long had tools for small-scale analysis of social networks - dozens or even hundreds of people. However, the explanations that showed social structure on a small scale - from Valdis Krebs's work on InFlow (SEE **RELEASE 1.0**, FEBRUARY 1996) to Harvard sociologist Mark Granovetter's work on the "weak ties" that hold a community together - didn't do a very good job of explaining social structure in groups of tens of thousands or more.

Because the "small-scale" and "Random" models emphasized the relatively homogeneous parts of human relations, neither accounted for the way a sparsely connected human network could exhibit "six degrees of separation," where any two people in even large groups can link to one another in a very short chain of acquaintances. (The phrase is from Stanley Milgram's research in the 1960s on how a message might pass from a sender in Omaha to a receiver in Boston if sender and receiver didn't know one another. He found that the average chain length - degrees of separation - was six.)

In 1998, Duncan Watts and Steve Strogatz proposed an alternative to the random network, called the Small Worlds model, which provides a good fit for real human networks. The essence of a Small World network is that it operates at two scales - local and "global" (which should more properly be called supra-local, indicating whatever scale is above local). In this two-scale model, the local scale is more tightly clustered than the supra-local scale. If the local scale is a working team, everyone will know everyone, while if the supra-local scale is the department that team is in, there may be some people who don't know one another, and the connections that are made will be weaker.

A Small World is one in which people know their co-workers and neighbors better than they know the people down the hall or down the block, but where larger agglomerations of people are held together by a handful of people who have connections that span (and therefore

link) clusters. Watts calls these people hubs, Malcolm Gladwell called them connectors in *The Tipping Point*, and their (usually unconscious) role is to bind the tighter local structures together into looser supra-local ones.

The research into Small Worlds dynamics is relatively new, but the explanatory power of the ideas is already astonishing, and resists easy summarization. Watts' recent book on the subject, *Six Degrees*, is well worth a read for anyone interested in the dynamics of social networks.

One of the surprises of the Small Worlds model is that as the system gets bigger, the difference between the best-connected individual hub and the average member of the Small World grows rather than shrinks. That is, in a Small World network, there will always be a "best-connected" hub, and the larger the system, the better connected that hub will be relative to everyone else. In large systems this connectedness follows a power law: i.e., the connector in the Nth position has 1/Nth the connectedness of the best-connected hub. The number-2 hub is half as well-connected, the number-10 hub is only a tenth as well connected, and so on.

In addition to Watts, a number of other people are researching network dynamics on the Internet, such as Albert-László Barabási of Notre-Dame, whose book *Linked* explores power-law dynamics in great detail. (Like Watt's *Six Degrees*, *Linked* is well-written, accessible to the layperson and quite wonderful.) Bernardo Huberman, Eytan Adar and Lada Adamic have all done significant work on the social network topologies that form on the Internet, first at Xerox PARC and now at HP Labs. This was the team that discovered that the Web is a small world network, and that on average, any two public websites are connected by four degrees of separation. They have recently been working on scouring the Web for data that will allow them to divine the social structure of groups.

Then there is the inimitable Valdis Krebs, who pioneered practical applications of Social Network Analysis (SNA) with his InFlow software. Krebs was one of the first people to try to take the work of mapping social networks out of a purely academic context and use it to drive business value. While his firm OrgNet is more focused on consulting than on shipping an SNA product (contrast Visible Path and Social Software, profiled later in this issue), his work is a nearly universal touchstone for people in the field.

Category: It's not *what* you know. . .

Social network analysis has seen explosive improvements in the last few years, as the availability of behavioral data from e-mail and the Web has provided a wealth of detailed data about social interactions. (SEE BOX, PAGE 21 AND CATAPHORA IN **RELEASE 1.0**, MARCH 2003.) The change might be likened to going from a snapshot to a mirror. Until recently, most SNA involved weeks or months of data gathering, followed by similarly long periods of analysis. This could produce a picture of a group from a moment in time, long after that time had passed. Now, with more interactions occurring electronically (and with better tools to record and analyze them), groups can see representations of social structure in near-real time; they can also see changes over time, and watch feedback from the changes. Snapshots are nice to have, of course, but it takes a mirror to make real-time adjustments with any confidence (just ask anyone getting her hair cut). Both Visible Path and Social Software intend to provide this kind of SNA-as-mirror function, providing businesses with quick pictures of complex social dynamics in ways that will help drive revenues.

Visible Path: Three degrees of connection

Ever since sociologist Stanley Milgram's original research into the length of the chains linking any two individuals, people have been fascinated with the topology of social connections. Recent research into the shape of large social networks by Duncan Watts and others (SEE PAGE 21), coupled with advances in information visualization (SEE **RELEASE 1.0**, SEPTEMBER 2002) have made it possible to derive and display the actual structure of a group of people.

Visible Path is built to leverage pull, guanxi, social capital, whuffie or whatever else you want to call it: Indeed, the core engine of its system is called the Relationship Capital Manager. Most businesses rely on social capital, but making it a manageable resource has been an elusive goal, in part because unlike financial capital, social capital does not exist separately from social networks – your relationships are a joint holding between you and the people you deal with. Antony Brydon, Visible Path's co-founder and CEO, likens our recent understanding of social structures to earlier advances in physics: "These complex group structures we live in are like social gravity. There are fundamental forces exerted by social networks that help explain phenomena like a marketing campaign's effectiveness or an organization's efficiency at handling change. Like gravity, these forces used to be mysterious. Now we're coming to understand them and be able to work with them."

Visible Path's executive team includes management from the Internet Underground Music Archive and sixdegrees.com, and they are applying the lessons they learned in the consumer sphere to the enterprise space. Stanley Wasserman, co-author of *Social Network Analysis: Methods and Applications* and a professor at the University of Illinois at Urbana-Champaign, is an advisor to the company. As you might expect, Visible Path's product is designed to show the user a social path between herself and other people: If you want to reach the head of purchasing at ACME Co., you can talk to Alice, who knows Bob, who knows the ACME purchasing employee. Visible Path takes the social data available in any organization, such as address books, calendars and most importantly e-mail headers, and turns it into a map showing the shortest path between you and the person you'd like to contact.

However, you don't immediately get to see the identities of the people on that path. As in the Milgram experiment, you may not know the people making the "small world" connection between you and your goal. Furthermore, they may not want to reveal that they know the target individual or someone else along the path. In fact, they may not particularly want to recommend you to that person. The canonical problem with CRM solutions is the unfounded assumption that everyone will be eager to share their contacts. Visible Path makes a different assumption. "We knew that any system that required full disclosure of contacts would fail, so we designed Visible Path to be opt-in for various levels of disclosure. Whatever data you put in the system can be anonymized, so that the software may show one or even several paths, but it won't disclose the names of the people in between without their permission." Note that "levels" of disclosure is a polite version of: "You can reveal my name to Juan, but not to Alice."

It works like this: If you want to know the best way to get to the aforementioned ACME executive (let's call her Carolyn), you would put her name into Visible Path. The software would then look at any contacts you had listed, and for anyone who'd listed Carolyn. A match would mean one degree of separation: Someone you know knows her. The system would also look for two- and three-link paths. Each node on that path is presented as an icon with only one piece of information: This person works at your company, this person works in the target company, or this person works elsewhere. Armed with that information, Visible Path would let you send mail to any of the people in your path, asking for an introduction. You wouldn't see who was getting the mail, but the recipient of that mail would see that it was from you. At that point, the recipient can do one of two things – delete the mail and do nothing,

VISIBLE PATH INFO
Headquarters: New York, NY
Founded: October 2002
Employees: 10
Funding: undisclosed amount from angels
Key metric: 36 of 40 sales VPs interviewed expressed interest
URL: http://www.visiblepath.com

or broker the introduction. In addition to this pull model, Visible Path can also automatically push leads to salespeople based on the strength of their connection to the lead, which saves sales reps from having to collaborate to pursue leads (an activity for which they have a noted aversion). Thus, in addition to showing the links, VP also does most of the work. It may sound trivial, but it makes it easy for people to do favors, and that's a big win in the time-short, task-crowded business world. . .

. . .and especially in the world of sales, the target market for the initial iteration of Visible Path. Says Brydon, "Our long-term goal is to be a general platform for integrating social network analysis into every aspect of business, but we chose to focus on sales up front. When we were interviewing prospective clients, it was sales and biz dev who told us we could solve a problem that every professional has." This means highlighting short chains of strong links. Although the idea of "six degrees" is captivating, in practice even four degrees – my colleague's friend's wife's boss – is too tenuous.

Social-Software: Identity is a two-way street

Social-Software is an early-stage company building software to help users manage their social environment. Corbin de Rubertis, Social-Software's Paris-based founder and CEO and the former VP of eBusiness at Novell, designed the software to help users traverse and manage the large number of relationships we all have to manage every day. "A key problem in most groupware," he says, "is that as the size of the group grows to even a few hundred, the software is no longer adequate to help you manage it. People need a better way to organize and traverse large lists of their relationships, in order to keep track of their social networks." De Rubertis believes that while the technological reality of social life is communication streams – e-mails, instant messages, phone calls and the like – the underlying user experience is one of relationships. When viewing e-mail from your boss, it's more important to know that it's from your boss than that it happens to be e-mail rather than, say, voice mail.

The idea behind the software is simple, and related to the kind of social data extraction behind both Visible Path and Valdis Krebs's InFlow: Monitor a user's behavior and extract social information from explicit things such as address books and e-mail headers, and from metadata such as the (relative) frequency of communications between two people vs. the communication patterns of each individual with other individuals. Next, build a database of those relationships. Finally, provide an interface to let the user query and annotate that social information, in order to help them manage those relationships.

THE 800-POUND GORILLAS

Social software is of course a key feature for established tech firms - and they have the advantage of being able to add it on to existing products and into existing installed bases. Microsoft made Outlook the default groupware client by making it the default e-mail application. AOL took in users by the millions by making e-mail and chat rooms easy to use (and later acquired ICQ, enabling it to dominate instant messaging). Yahoo! purchased eGroups and turned it into Yahoo! Groups. And even Google is getting into the act with its recent purchase of Pyra Labs, creators of Blogger, the original weblogging platform.

It is a curiosity of social software that the traditional categories of consumer and business use are blurred. The largest provider of IM services within the enterprise today is AOL, because IM started as a consumer application and was brought into businesses by the employees, instead of the IT department. The same pattern happened earlier with e-mail, with consumer ISPs often pressed into enterprise use by employees. It's happening again with weblogs and wikis. This brings about an odd competitive mix, where Microsoft Outlook and Yahoo! Groups are the de-facto competition for Lotus Notes and Groove - since the adoption and use of social software is often organic and bottom-up, there are other ways to get into an enterprise than having an enterprise sales team.

The big players present both significant risk and opportunity for startups in the social software space. Software almost always delivers positive returns to scale (it's easier to get bigger if you're already big). Because social software with few users doesn't deliver much value, the economies-of-scale equation is worse for startups in this area: Having one of the first spreadsheet programs is interesting, while having one of the first IM programs is not nearly as useful. (Whom would you talk to?)

Most of this market risk comes from Microsoft. Its enormous reach is doubly threatening to startups when coupled with Microsoft's interest in adding or improving social features: Outlook's calendar feature works because

it is integrated with e-mail; Windows XP does everything but bribe the users to get them to set up a Windows Instant Messaging client. Microsoft has several research efforts in this area, including Marc Smith's brilliant Netscan research, data-mining Usenet for social patterns, the youth-oriented 3° (3 Degrees) social computing tool, and its Polyarchy work on visualizing social hierarchies (SEE **RELEASE 1.0**, MARCH 2003).

An even bigger revenue threat than Microsoft may come not from a company but a movement - open source. Because this generation of social software tools is so simple, and because a common pattern of open-source development is to copy an existing commercial feature set, the presence of free alternatives to commercial social software will make pure product offerings difficult to make money on. Startups with social software offerings will likely turn to hosting, customization, and consulting contracts to replace revenue from simple product sales.

The two big service firms, IBM and HP, represent more opportunity than Microsoft, even though they compete more directly in the consulting area. Like Microsoft, both have serious research arms: IBM has both Lotus and its Collaborative User Experience Group in Cambridge, and HP has hired several researchers in network dynamics from PARC, including Bernardo Huberman (SEE **BOX**, PAGE 21). Though they both pose some of the same threat as Microsoft, both also have service arms. Having consultants who know how to install and maintain third-party software always makes a company more amenable to working with outside firms, especially firms that are too small to offer a global or 24/7 service component on their own.

The biggest opportunity for social software developers may be AOL, Yahoo! and Google. All of these firms have enormous user bases to support, and helping the users to create social value for one another at a low cost could be a key business driver. They may continue their habit of acquiring startups that provide otherwise hard-to-monetize value for users.

Once the basic data about your social universe has been extracted, you begin to get a picture of your own social identity - or identities, which often vary according to whom you are interacting with. De Rubertis says, "We presuppose a richer definition of both identities and relationships than is typically captured in things like e-mail addresses. Our goal is to extract the more nuanced aspects of relationships and iden-

ties, and to make those available to the user.” Once your basic profile exists within the system, you can add detail about your own identity in a process he likens to preparing a resume: all the schools you attended, all the businesses you worked at, all the places you lived. Like the key aspects of weblog publishing, Social-Software makes human identity and time the major axes of organization of the data in the system.

You can also annotate other people’s identities once they have been discovered by the system – the people you trade e-mail or IMs with – though De Rubertis hopes that the system grows quickly enough among clustered groups of users that they can begin to syndicate their own descriptions of themselves to other people, like a particularly expressive vCard. (Of course, someone’s own view of himself may not be the only view you want. It’s unlikely that Bill Bennett, author of *The Book of Virtues*, would list a gambling habit in his profile, for example.)

Overcoming the inertia that has hampered the vCard – a technology that will only really work after millions adopt it – is a significant problem. On installation, Social-Software creates a picture of the user’s social universe, updated daily, as a way of luring the user into the system. The software will show you who’s new in your network, as well as anything that may have changed. If two people you know suddenly show up on the same Cc: line, the interface will show you that they now share at least that connection. (The process is analogous to the social version of Googlealert.com, the service that alerts you when anything on the front page of a particular Google search changes.) This is like a daily dashboard of your contacts, expressed as relationships rather than communications streams. (It’s a daily dashboard rather than a real-time one, because social network analyses are notoriously complex. The software is currently designed to use the local PC’s unused cycles to gather and interpret relationship data, and De Rubertis is looking at peer-to-peer and server-based versions as alternate ways of deploying the necessary horsepower.)

SOCIAL-SOFTWARE INFO
Headquarters: San Francisco, CA
Founded: November 2002
Employees: 6
Funding: undisclosed amount from angels
URL: http://www.social-software.com

The real value, however, is in being able to query the database to get new information, or to use it as a co-browser, pulling and analyzing profile data directly from Web pages as a user browses. Social-Software runs on Eric Schmidt’s adopted dictum: “The antidote to bad information is more information.” (De Rubertis used to work for Schmidt at Novell.) De Rubertis says, “We’d like all our data to be as structured as an address book, but it isn’t. Google has taught us all the value of approximate answers. We can extract those approximate answers from publicly available data sources.” A weakness of social networking tools such as Ryze and Friendster is

that they know only about other people in the system. De Rubertis suggests that you can query the Social-Software database for things like, “Whom do I know that knows Steve Case?” and have it assemble an answer in part on your local data and in part on external data.

Social-Software is self-funded (though De Rubertis is raising an angel round) and looking for beta-testing clients. “There’s still quite a bit to do on the server-hosted version, but we’ll launch the Web version, free to download and use on your own, by the second half of this year.”

The bet here is much the same as the one Visible Path is making, namely that producing useful details about a firm’s social network, both within and outside its four walls, will be worth paying for. “While social network analysis is interesting,” says De Rubertis, “the results are extremely arcane and very difficult to convey to a lay audience. You need to make it simple enough for the users to create value on their own.”

Social Software: Take the Bad with the Good

The most radical change social software brings about is the decoupling of groups from needing to meet in the same place and time, with both positive and negative effects. Decoupling geography from conversation makes global collaboration possible, but the lack of fact-to-face connection can remove or hide critical social cues. Asynchrony makes information coordination problems much simpler, but lack of context can make people feel less comfortable about decisions than they would be in a physical gathering. Machine-readable data makes archiving, searching and repackaging trivial, but these capabilities can backfire in the case of privacy spills and over-wide distribution.

These different social patterns mean that designing and deploying social software can’t be done in the same way as for, say, a word processor. Using social software is not better – or even always worse – than attending face-to-face meetings; it is different. Indeed, for some functions, such as brainstorming or post-mortem analysis, there’s evidence that having the participants separated in space may help avoid groupthink and allow the less assertive participants to get a hearing. Given that social software is not a pure replacement for real-world meetings, the design problem becomes one of deploying social software to augment, rather than merely replace, existing collaborative systems. The companies that want to make money

building, deploying or supporting social software have to take account of human factors, in particular the inevitability of failure in some social situations, in ways that vendors of single-user software do not.

It's safe to predict a revolution in our understanding of group dynamics; indeed, thanks to work by theorists such as Watts, Wasserman, Barabási and others, it's already underway. It's also safe to predict a parallel revolution in the way we use technological tools to mediate those group dynamics, to amplify or alter certain behaviors. It's even safe to predict revolution in the way business conducts its

affairs: As we know from the impact of e-mail and IM and are seeing now with weblogs, when you change the ways individuals communicate, the effects are quickly felt by businesses.

COMING SOON

- Weblogs and publishing.
 - Enum and registries.
 - Reputation systems.
 - Location-based services.
 - Non-homeland security.
- And much more. . . (If you know of any good examples of the categories listed above, please let us know.)

Sometimes the change comes from the outside and moves in: Robert Scoble, a former NEC engineer, uses his weblog to lay out, in damning detail, Microsoft's poor handling of community relations. Fast-forward six months and Microsoft has hired him in part to help address the some of the problems he described from the outside. Sometimes the change comes from the inside and moves out: Joel Spolsky, CEO of Fog Creek Software, uses his blog to hash out in public ideas for software his company is working on.

Sometimes the change is from the top down: Ray Ozzie, founder and CEO of Groove, recognizes that employee blogging is an inevitability, so he sets up his own blog in order to understand the allure and lead by example. And sometimes – often, in fact – the change is from the bottom up. In any tech-savvy company of more than a dozen employees, it's a safe bet that at least one of them has started a blog, and may well be using it to discuss his employer.

Because many of the current tools are easy to configure, easy to use, and serverless, individuals can easily adopt them without official involvement or approval. They are entering the corporate environment through the side door, one user at a time. There's no guarantee things will stay that way, however, and good reason to think they won't. Ten years ago, there was a debate about whether e-mail was ready for business, but once employees answered the question in the affirmative, the IT department got involved and asserted its control. As Greif notes, "Even technologies that are brought in by individuals and that seem to have flexibility can be reclaimed by IT if they become popular enough to cause infrastructure 'issues.' E-mail is an

Resources & Contact Information

Duncan Watts, **Columbia University**, 1 (212) 854-4343; fax, 1 (212) 854-2963; djw24@columbia.edu
Phil Clevenger, **Cooperating Systems Inc**, 1 (805) 886-2797; fax, 1 (805) 564-7188; phil@cooperatingsystems.com
Kai Gradert, **Cooperating Systems Inc**, 1 (323) 314-4485; fax, 1 (805) 564-7188; kai@cooperatingsystems.com
Ward Cunningham, **Cunningham and Cunningham**, 1 (503) 245-5633; ward@c2.com
Julio Estrada, **Kubi Software**, 1 (781) 259-7900; fax, 1 (781) 259-8963; info@kubisoftware.com
Bernardo Huberman, **HP Labs**, 1 (650) 857-5318; fax, 1 (650) 813-3706; huberman@hpl.hp.com
Irene Greif, **IBM**, 1 (617) 693-5789; fax, 1 (617) 693-5551; igreif@us.ibm.com
Marc Smith, **Microsoft**, 1 (425) 936-6896; masmith@microsoft.com
Albert-László Barabási, **University of Notre Dame**, 1 (574) 631-5767; fax, 1 (574) 631-5952; alb@nd.edu
Valdis Krebs, **OrgNet**, 1 (440) 331-1222; fax, 1 (440) 808-0883; valdis@orgnet.com
Steve Crocker, **Shinkuro**, 1 (202) 824-0708; fax, 1 (202) 478-1723; steve@shinkuro.com
Corbin De Rubertis, **Social-Software**, corbin@social-software.com
Ross Mayfield, **Socialtext**, 1 (650) 323-0800; fax, 1 (650) 323-0801; ross.mayfield@socialtext.com
Greg Lloyd, Traction Software, 1 (401) 528-1145; fax, 1 (443) 331-2549; grl@tractionsoftware.com
Stanley Wasserman; **University of Illinois, Urbana-Champaign**, 1 (217) 244-6905; fax, 1 (217) 244-8371; stan-wass@uiuc.edu
Antony Brydon, **Visible Path**, 1 (877) 847-7284; fax, 1 (212) 202-4213; abrydon@visiblepath.com

(resources on next page)

IT-deployed system, but for the most part is deployed in a way that lets it support very flexible peer-to-peer communication and networking. I would claim that most team space products have not. People have to request team spaces rather than create them spontaneously.” So the pendulum can swing.

And then there’s the one complexity that never goes away: people. Because solving one problem creates unexpected new side-effects, there will never be any perfect social software outside the world of the “Matrix.” The best any new generation of social software can do is to make things better than they were.

All these factors make it difficult to predict commercial ramifications. Several things make it hard for a vendor to capture as revenue the obvious value social software creates: It has always been difficult to charge money for the creation of social value, especially online; e-mail and IM offer a free alternative (from the user’s point of view) to any product that costs money; corporations require security and control; and open source and BigCo bundling are a threat to quickly offer the same features for free. Despite these challenges to commercial success, this generation of social software will certainly affect the businesses that adopt it – or whose employees sneak it in. ■ R 1.0

Resources & Contact Information

Weblogging platforms:

Blogger: <http://www.blogger.com> (Look at the "Fresh Blogs" column for examples)

LiveJournal: <http://livejournal.com> (Click "Random" under "Find Users" for examples.)

Radio Userland: <http://radio.userland.com> (The weblogs themselves are listed at <http://www.movabletype.org>)

Movable Type: <http://www.movabletype.org> (Look at "Recently Updated" for examples.)

Sample weblogs:

Joi Ito: <http://joi.ito.com>

Robert Scoble: <http://radio.weblogs.com/0001011/>

Joel Spolsky: <http://www.joelonsoftware.com>

Dave Winer: <http://www.scripting.com>

Group-run weblog on social software: <http://www.corante.com/many/> (Clay Shirky is a contributor.)

N.Z. Bear's cross-blog debate:

http://www.truthlaidbear.com/archives/2003/02/10/crossblog_iraq_debate_the_questions.php

Individual broadcasts to the world (observations of an Oakland-based Web developer):

<http://www.littleyellowdifferent.com>

Serious media outlet (law professor with libertarian/right views and monthly traffic of over a million viewers):

<http://www.instapundit.com>

Wiki information:

Ward Cunningham's original wiki: <http://www.c2.com/cgi/wiki/>

List of wiki hosting options: <http://c2.com/cgi/wiki?WikiFarms>

Other software:

Netscan: <http://netscan.research.microsoft.com>

3°: <http://www.threedegrees.com>

Hydra: <http://hydra.globase.org/>

"Happenings": <http://socialtext-com.istori.com/weblog/>

Further reading:

Albert-László Barabási, **Linked**, (Perseus Publishing, May 2002)

Duncan J. Watts, **Six Degrees: The Science of a Connected Age**, (W.W. Norton & Company, February 2003)

Stanley Wasserman, Katherine Faust, Dawn Iacobucci, **Social Network Analysis : Methods and Applications**,
(Cambridge University Press, November 1994)

David Weinberger, **Small Pieces Loosely Joined: A Unified Theory of the Web**, (Perseus Publishing, March 2002)

Malcolm Gladwell, **The Tipping Point: How Little Things Can Make a Big Difference**, (Little Brown & Company,
February 2000)

Calendar of High-Tech Events

- JUNE 7-8 **FreeNetworks Conference 2003** – Las Vegas, NV. Meet the people building community wireless networks across the world. Speakers include PC Forum speaker Tim O'Reilly and Cory Doctorow. For more information or to register, visit the website. con.freenetworks.org
- JUNE 9-11 **Bled eCommerce Conference** – Bled, Slovenia. Business, government and technologists from around the world gather in Bled to discuss research and business issues surrounding ecommerce. Register online or contact Kristina Bogataj, +386 (4) 237-4291; fax, +386 (4) 237-4365; email, Kristina.Bogataj@FOV.uni-mb.si. ecom.fov.uni-mb.si/Bled2003
- JUNE 11-13 **RFID Journal Live!** – Chicago, IL. Organized by RFID Journal, this conference will explore how RFID will impact your business, with early adopters sharing their experiences and lessons learned. Register online or call 1 (510) 832-1500. For more information contact Daniella Seghieri, Daniella@phocas-partners.com. www.rfidjournallive.com **E**
- JUNE 11-13 **TedMed3** – Philadelphia, PA. Discover how technology can help you achieve a healthier life. Imagine! Register online or call (401) 848-2299; e-mail, wurmanrs@aol.com. www.tedmed.com
- JUNE 18-20 **CeBIT** – New York, NY. Europe's biggest technology trade show comes to America. For information about registering or exhibiting visit the website. www.cebit-america.com
- JUNE 19-20 **Free/Open Source Software Conference** – Cambridge, MA. Explore new models for software development...and the OS community's impact on sociology, economics and management. For more information, visit the website or email osconf@mit.edu. opensource.mit.edu/conference.html
- JUNE 23-27 **IPv6 Global Summit** – San Diego, CA. The not-to-miss event for the IPv6 set. Register online or contact Alex Lightman, alex@charmed.com. www.usipv6.com
- JUNE 23-27 **ATPN 2003** – Eindhoven, The Netherlands There's much to be learned about networked systems from biology. Discover the wisdom of Petri Nets at the International Conference on Application and Theory of Petri Nets, in its 24th year. Register online or e-mail atpn2003@tue.nl. www.tue.nl/atpn2003
- JUNE 26-27 **UpStart Europe 2003** – London, UK. In its fourth year, UpStart will give technology entrepreneurs in Europe that get-up-and-go feeling. Register online or call +31 (20) 462-1983. www.tornado-insider.com/upstarteurope
- JULY 7-11 **O'Reilly Open Source Convention** – Portland, OR. A central gathering place for the open source community. Register online or call Linda Holder, (800) 998-9938 or (707) 827-7000 (outside the US); fax, (707) 829-1342; lholder@oreilly.com. conferences.oreillynet.com/os2003/

E Events Esther plans to attend.

Lack of a symbol is no indication of lack of merit. The full, current calendar is available on our website, www.edventure.com. Please contact Christina Koukkos (christina@edventure.com) to let us know about other events we should include.

The conversation never stops! Subscribe to our free e-mail newsletter, *The conversation continues*, for thought-provoking analysis from our editors, along with commentary from our highly intelligent readers. Sign up at <http://www.edventure.com/conversation/join.cfm>.

Release 1.0 Subscription Form

Complete this form and join the other industry executives who regularly rely on *Release 1.0* to stay ahead of the headlines. Or if you wish, you can also subscribe online at www.release1-0.com.

Your annual *Release 1.0* subscription costs **\$795** per year (**\$850** outside the US, Canada and Mexico), and includes both the print and electronic versions of 11 monthly issues; 25% off the cover price when you order from our online archives; a *Release 1.0* binder; the bound transcript of this year's PC Forum (a \$300 value) and an invitation to next year's PC Forum.

NAME _____
TITLE _____ COMPANY _____
ADDRESS _____
CITY _____ STATE _____ ZIP _____ COUNTRY _____
TELEPHONE _____ FAX _____
E-MAIL* _____ URL _____
*personal e-mail address required for electronic access.

My colleagues should read *Release 1.0*, too!
Send me information about multiple copy subscriptions and electronic site licenses.

Check enclosed Charge my (circle one): AMERICAN EXPRESS MASTER CARD VISA

CARD NUMBER _____ EXPIRATION DATE _____
NAME AND BILLING ADDRESS _____
SIGNATURE _____

Please fax this form to Natasha Felshman at 1 (212) 924-0240.

Payment must be included with this form. Your satisfaction is guaranteed or your money back.

If you wish to pay by check, please mail this form with payment to: EDventure Holdings, 104 Fifth Avenue, 20th Floor, New York, NY 10011, USA. If you have any questions, please call us at 1 (212) 924-8800; e-mail us@edventure.com; www.edventure.com.