CHAPTER 12

Mind to Mind

I have a dream for the Web . . . and it has two parts.

In the first part, the Web becomes a much more powerful means for collaboration between people. I have always imagined the information space as something to which everyone has immediate and intuitive access, and not just to browse, but to create. The initial *WorldWideWeb* program opened with an almost blank page, ready for the jottings of the user. Robert Cailliau and I had a great time with it, not because we were looking at a lot of stuff, but because we were writing and sharing our ideas. Furthermore, the dream of people-to-people communication through shared knowledge must be possible for groups of all sizes, interacting electronically with as much ease as they do now in person.

In the second part of the dream, collaborations extend to computers. Machines become capable of analyzing all the data on the Web—the content, links, and transactions between people and computers. A "Semantic Web," which should make this possible,

has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy, and our daily lives will be handled by machines talking to machines, leaving humans to provide the inspiration and intuition. The intelligent "agents" people have touted for ages will finally materialize. This machine-understandable Web will come about through the implementation of a series of technical advances and social agreements that are now beginning (and which I describe in the next chapter).

Once the two-part dream is reached, the Web will be a place where the whim of a human being and the reasoning of a machine coexist in an ideal, powerful mixture.

Realizing the dream will require a lot of nitty-gritty work. The Web is far from "done." It is in only a jumbled state of construction, and no matter how grand the dream, it has to be engineered piece by piece, with many of the pieces far from glamorous.

It is much easier to imagine and understand a more enlightened, powerful Web if we break free of some of the world's current assumptions about how we use computers. When I want to interact with a computer, I have to wait several minutes after turning it on before it is ready to converse. This is absurd. These machines are supposed to be there for us, not the other way around. So let's begin our thinking about a new world by imagining one in which a computer screen is available whenever we want it.

In the same spirit, we should jettison our assumptions about Internet access. Why should we have to wait while a computer connects to the Internet by making a phone call? The Internet isn't designed to be like that. It is made so that, at any time, a little postcardlike packet of a few hundred characters could be dropped into it by one computer, and in a fraction of a second be at its destination on the other side of the world. That is why clicking on an icon can take us very quickly to a Web site. The bother of having to make a phone call wrecks the idea of instant availability.

An essential goal for the telecommunications industry (and regulatory authorities) should be connecting everyone with *permanent* access. The problem till now has not been technology, but rather regulations that control what telephone companies can charge for access, and the lack of agreement about how other companies that might want to provide Internet access can lease the copper wire that goes to every home. With some wiser regulation, in some cases spurred on by competition from cable companies that lay their own cables to people's doors, before too long I should be able to walk up to a screen, see it quickly glow with my home page on it, and follow a link immediately. This simple difference in timing will dramatically change the way we use computers, making the experience more like getting out a pen rather than getting out a lawnmower. Computers will be there when we suddenly have an idea, allowing us to capture it and preventing the world from losing it.

Let's clear our minds about what we will see on these wonderful new computers. Today there is a desktop with various folders and "applications." One of these applications is a Web browser. In this scheme, my entire screen is taken up by my local computer, while all the information in the rest of the accessible world is relegated to a small area or icon within it. This is inside out.

The job of computers and networks is to get out of the way, to not be seen. This means that the appearance of the information and the tools one uses to access it should be independent of where the information is stored—the concept of *location independence*. Whether they are hypertext pages or folders, both valid genres of information management, they should look and feel the same wherever they physically happen to be. Filenames should disappear; they should become merely another form of URI. Then people should cease to be aware of URIs, seeing only hypertext links. The technology should be transparent, so we interact with it intuitively.

The next step would be protocol independence. Right now, every time I write something with a computer, I have to choose whether to open the "electronic mail" application or the "net news" application or the "Web editor" application. The mail, news, and Web systems use different protocols between computers, and effectively, I am being asked to select which protocol to use. The computer should figure this out by itself.

Location independence and protocol independence would be very simple if all the software on a computer were being rewritten from scratch. Unfortunately, it isn't. The required change to the modular design of operating systems and applications would be significant. Indeed, whether or not the terms *operating system* and *application* would survive is not clear. But since software engineers are very inventive, and the stakes—an intuitive interface—are high, I am optimistic.

As we look at the way a person uses the Web, it is simplest to improve the reception of information by adding new forms of graphics and multimedia. It is more difficult to imagine how best to allow a person to interact with the information, to create and modify it. Harder still is imagining how this computer screen can be used to allow one person to interact as one of many people interacting as a group. This is the order in which development has occurred to date, and will occur in the future.

The XML revolution, mentioned in chapter 9, that has taken place over the last few years and is now reaching the mainstream has provided a solid foundation for much of the new design inside and outside the consortium. Even though the computer markup languages for hypertext and graphics are designed for presenting text and images to people, and data languages are designed to be processed by machines, they share a need for a common, structured format. XML is it.

XML is both a boon and a threat to the Web dream. The great thing is that it stems the tide of information loss. It allows anyone to create any kind of tag that can capture the intent of a piece of

information. For example, the minutes of a meeting may contain an "action item." XML allows the person taking the minutes to make a new document type that includes <*action*> as a new tag. If the minutes are recorded in HTML, this might be lost, because HTML's general set of tags don't include <*action*> and the person taking the minutes can't create one. An XML document is typically richer: The information it contains is more well defined.

This will allow such things as spreadsheets, calendar files, e-mail address books, and bank statements that have not used interoperable standard formats to have them developed quickly, dramatically increasing the interoperability in, for example, typical office documents. This is the primary excitement behind the XML revolution: avoiding the information lost when such documents are translated into HTML and thereby lose their ability to be understood as spreadsheets, calendars, bank statements, or whatever.

The threat is that when a company introduces a new document type, no one else will understand it. XML makes it easy for everyone to create their own tags or entire markup languages. We might therefore see an end to the idyllic situation that has prevailed thus far on the Web—the predominance of HTML, which has helped all of us share documents easily. Can it be that, a decade into the Web's existence, XML will give us a freedom that forcibly leads us back toward myriad incompatible languages? This is indeed a serious possibility, but one that has been anticipated.

The extensible *X* in *XML* means anyone can invent new tags, but they can't add them to someone else's tags. An XML document can be made of a mixture of tags from more than one *namespace*, but each namespace is identified by a URI. Thus any XML document is completely defined using the Web. This is a huge step forward from the old HTML days in which anyone could make up their own version of what <*table*> meant, for example, with no ambiguity. The XML namespaces change the rules of

technology evolution by making every step, whether open or proprietary, well defined.

It is important to remember that XML does not replace HTML. It replaces the underlying SGML on which HTML was built. HTML can now be written as XML. In fact, it is possible to create a valid XML document that will also work with old HTML browsers. (The specification for doing this is XHTML.)

When I proposed the Web in 1989, the driving force I had in mind was communication through shared knowledge, and the driving "market" for it was collaboration among people at work and at home. By building a hypertext Web, a group of people of whatever size could easily express themselves, quickly acquire and convey knowledge, overcome misunderstandings, and reduce duplication of effort. This would give people in a group a new power to build something together.

People would also have a running model of their plans and reasoning. A web of knowledge linked through hypertext would contain a snapshot of their shared understanding. When new people joined a group they would have the legacy of decisions and reasons available for inspection. When people left the group their work would already have been captured and integrated. As an exciting bonus, machine analysis of the web of knowledge could perhaps allow the participants to draw conclusions about management and organization of their collective activity that they would not otherwise have elucidated.

The intention was that the Web be used as a personal information system, and a group tool on all scales, from the team of two creating a flyer for the local elementary school play to the world population deciding on ecological issues.

I also wanted the Web to be used just as much "internally" as externally. Even though most of the first ten servers, like the one at CERN or SLAC, would be called *intranet* servers today, organizations and families are just beginning to see the power the Web

can bring inside their walls. Although it takes a little work to set up the access control for a corporate or family intranet, once that has been done the Web's usefulness is accelerated, because the participants share a level of trust. This encourages more spontaneous and direct communication.

To be able to really work together on the Web, we need much better tools: better formats for presenting information to the user; more intuitive interfaces for editing and changing information; seamless integration of other tools, such as chat rooms, and audio- and videoconferencing, with Web editing. We need the ability to store on one server an annotation about a Web page on another; simple access controls for group membership, and for tracking changes to documents. While some of this work involves leading-edge research, a lot of it consists of trying to adapt existing computer systems to the global hypertext world.

For people to share knowledge, the Web must be a universal space across which all hypertext links can travel. I spend a good deal of my life defending this core property in one way or another.

Universality must exist along several dimensions. To start with, we must be able to interlink any documents—from drafts to highly polished works. Information is often lost within an organization when a "final document" of some kind is created at the end of an endeavor. Often, everything from the minutes of meetings to background research vanishes, and the reasoning that brought the group to its endpoint is lost. It might actually still exist on some disk somewhere, but it is effectively useless because the finished document doesn't link to it. What's more, different social and practical systems isolate documents of different levels from each other: We don't insert random notes into finished books, but why not, if they are relevant and insightful? At the consortium today, no one can mention a document in a meeting unless they can give a URI for it. Our policy is "If it isn't on the Web, it doesn't exist," and the cry often heard when a new

idea is presented is "Stick it in Team Space!"—a directory for confidentially saving documents not otherwise on the Web. All mail is instantly archived to the Web with a persistent URI. It is already hard to imagine how it could have been any other way. The Web of work and play must be able to intertwine half-baked and fully baked ideas, and Web technology must support this.

Another dimension critical to universality is the ability to link local material to global. When an endeavor is put together that involves groups of different scales—whether a software engineering project such as mine at CERN, or an elementary school education project that is part of a town initiative and uses federal funds—information has to come from many levels and has to be cross-linked.

Similarly, universality must exist across the spectrum of cost and intention. People and organizations have different motivations for putting things on the Web: for their own benefit, commercial gain, the good of society, or whatever. For an information system to be universal, it can't discriminate between these. The Web must include information that is free, very expensive, and every level in between. It must allow all the different interest groups to put together all manner of pricing and licensing and incentive systems . . . and always, of course, allow the user to "just say no."

The reason we need universality on all these levels is that that's how people operate in the real world. If the World Wide Web is to represent and support the web of life, it has to enable us to operate in different ways with different groups of different sizes and scopes at different places every day: our homes, offices, schools, churches, towns, states, countries, and cultures. It must also transcend levels, because creative people are always crossing boundaries. That is how we solve problems and innovate.

Information must be able to cross social boundaries, too. Our family life is influenced by work. Our existence in one group affects that in another. Values and actions are fed by all the ideas

from these different areas. By connecting across groups, people also provide organization and consistency to the world. It is unusual for an individual to support environmental policies on a global level but then plan to dump chemicals into the local river.

My original vision for a universal Web was as an armchair aid to help people do things in the web of real life. It would be a mirror, reflecting reports or conversations or art and mapping social interactions. But more and more, the mirror model is wrong, because interaction is taking place primarily on the Web. People are using the Web to build things they have not built or written or drawn or communicated anywhere else. As the Web becomes a primary space for much activity, we have to be careful that it allows for a just and fair society. The Web must allow equal access to those in different economic and political situations; those who have physical or cognitive disabilities; those of different cultures; and those who use different languages with different characters that read in different directions across a page.

The simplest factor controlling the Web as a medium for communication between people is the power of the data formats used to represent hypertext, graphics, and other media. Under pressure because of their direct visibility and impact on the user's experience, these have advanced relatively rapidly, because each medium has been essentially independent of the others.

One might have expected that graphics formats would have been standardized long ago, but the Web introduced new stresses that are forcing quite an evolution. Marc Andreessen gave browsers the ability to display graphics right inside a document, instead of relegating them to a separate window. He happened to pick the Graphic Interchange Format (GIF) defined by CompuServe. Soon, people also started using the standard JPEG (Joint Photographic Experts Group) format for photographs. These two formats reigned supreme until Unisys announced that it had ended up being the owner of a patent on the compression technology

used to make GIF images and that they would be charging license fees. A small group of enthusiasts proposed an alternative, Portable Network Graphics (PNG), based on an open compression technology, and generally superior to GIF. The consortium members agreed to endorse PNG as a W3C recommendation.

The recent moves to put the Web on everything from televisions to mobile phone screens have made the need for device dependence very clear. This has prompted even newer graphics formats that are more capable of displaying an image on screens of different sizes and technologies. Both JPEG and PNG describe a picture in terms of the square grid of *pixels* that make up a computer screen. The consortium is developing a new format for drawings that will describe them as abstract shapes, leaving the browser free to fill in the pixels in such as way that the image can be shown with optimal clarity on a wristwatch or a drive-in movie screen. The format, called *scalable vector graphics*, is based on XML. It will also dramatically speed up the delivery of documents containing drawings, which will open the door to all sorts of new ways of interacting between a person and a Web site. And because it is in XML, it will be easy for beginners to read and write. We may soon see all kinds of simple animated graphical interfaces.

Virtual Reality Modeling Language (VRML) is another pillar, being created for three-dimensional scenes. I expected 3D to really take off, and still don't quite understand why it hasn't. Sending the details of a 3D scene takes relatively few bytes compared, for example, with video. It does require the user to have a fast computer, to manipulate the scene as the user moves around it. Perhaps the power of the average processor just isn't high enough yet.

Integrating many different text, image, audio, and video media into one Web page or show will be greatly helped by the Synchronized Multimedia Integration Language (SMIL; "smile"). SMIL will make seamless coordination simple, even for authors with limited Web design experience. The notorious Clinton tapes, relayed over the Web in windows with mixtures of graphics, text,

and video, were if nothing else a launch for SMIL. The language can also effectively save bandwidth. Often a TV signal—say, a news broadcast—has a talking head that takes up maybe a quarter of the screen, a still image or map in the background, and perhaps a caption, not to mention basketball scores scrolling across the bottom of the screen. Transmitting all that as video data takes a lot of bandwidth. SMIL allows the relatively small amount of data about images that are actually moving to be sent as video, and integrated with the still images that are transmitted to the viewer's screen in ways that require much less bandwidth.

Running through all the work on hypertext, graphics, and multimedia languages are concerns about access for all, independent of culture, language, and disability. The consortium's Web Accessibility Initiative brings together people from industry, disability organizations, government, and research labs to devise protocols and software that can make the Web accessible to people with visual, hearing, physical, and cognitive or neurological disabilities. The work ranges widely, from review of W3C technologies to ensure that they support accessibility to development of accessibility guidelines for Web sites, browsers, and authoring tools, and development of tools to evaluate accessibility. Much of this works only when those building Web sites have taken a little care about how they have done it. The disability and technical communities got together to produce a set of guidelines about the most effective and practical steps to take: recommended reading for webmasters.

The consortium also has an internationalization activity that checks that new specifications will work in different alphabets, whether they are Eastern or Western, read right to left, left to right, or up and down. Conversions can get complicated, but the computer industry is making energetic efforts to extend operating systems to support the display of all kinds of written scripts, including Arabic, Hindi, Korean, Chinese, Japanese, Thai, and Hebrew. HTML 4.0 already provides a number of internationalization

features, including the ability to mark text as to which language it is in, and to order text from right to left.

The primary principle behind device independence, and accessibility, is the separation of *form* from *content*. When the significance of a document is stored separately from the way it should be displayed, device independence and accessibility become much easier to maintain. Much of this is achieved with a *style sheet*—a set of instructions on how to present or transform a printed page. Håkon Lie, who worked with me at CERN and later at the consortium, led the development of Cascading Style Sheets (CSS) to make this possible. A new, related language with different capabilities, XSL, is also in the works. There is even an "aural" style-sheet language, part of CSS2, to explain to a browser how a Web page should sound.

The growing list of graphics formats relate primarily to static displays. But some people feel a Web page isn't sufficiently exciting unless it moves. At a minimum, they want the page to change as a user interacts with it. Pop-up balloons and menus, and forms that fill themselves in, are simple examples we find today on the Web. These work because a small program, or *script*, is loaded with the page. It operates the page like the hand inside a puppet, in response to the user's actions. This has created a crisis in interoperability, however, because the connection between the script and the Web page, the hand and the puppet, is not standard for different kinds of style sheets. To fix this, the consortium is working on a Document Object Model (DOM), a set of standards for that interface. Unfortunately, it is much more difficult to make these animated pages accessible to voice browsers and screen readers. On the positive side, the DOM interface should provide a powerful way for accessibility tools such as document readers to access the document structure within a browser.

The media may portray the Web as a wonderful, interactive place where we have limitless choice because don't have to take what

the TV producer has decided we should see next. But my definition of interactive includes not just the ability to choose, but also the ability to create. We ought to be able not only to find any kind of document on the Web, but also to create any kind of document, easily. We should be able not only to follow links, but to create them—between all sorts of media. We should be able not only to interact with other people, but to create with other people. *Intercreativity* is the process of making things or solving problems together. If *interactivity* is not just sitting there passively in front of a display screen, then *intercreativity* is not just sitting there in front of something "interactive."

With all this work in the presentation of content, we still have really addressed only the reading of information, not the writing of it. There is little to help the Web be used as a collaborative meeting place. Realizing this early on, the consortium held a workshop to find out what was needed. The result was a long shopping list of capabilities, things like strong authentication of group members, good hypertext editors, annotation systems (similar to the little yellow paper sticky notes), and tools for procedures such as online voting and review.

Some of the results have been satisfying. SMIL was one, integrating various media and possibly allowing a real-time collaborative environment, a virtual meeting room, to be constructed. Others are still in the wings. A long-standing goal of mine had been to find an intuitive browser that also, like my *WorldWideWeb,* allows editing. A few such browser/editors had been made, such as AOLpress, but none were currently supported as commercial products. Few items on the wish list for collaborative tools had been achieved. At the consortium we wondered what was wrong. Did people not want these tools? Were developers unable to visualize them? Why had years of preaching and spec writing and encouragement got hardly anywhere?

I became more and more convinced that the only way to find out what was holding back the development of collaborative tools

was to try to develop them ourselves. Our policy had always been that we would use whatever commercial tools were available to get our own work done. At a consortium team retreat in Cambridge, I suggested we start trying all the experimental solutions being tinkered with in the community, and even develop them further. Perhaps then we would stumble upon the real problems, showing the way toward solutions.

We concluded that to do this, we needed a nucleus of people who would try various new collaboration technologies, just to see what happened. They would help the entire consortium staff become early adopters of experimental software. This new policy, which we called Live Early Adoption and Demonstration (not coincidentally, LEAD), meant that we entitled ourselves to eat our own dog food, as far as our very limited resources would allow. It meant that we'd be testing new protocols not on their own, but in the context of our actual, daily work. It also meant that, with only a handful of programmers, we would be trying to maintain the reliability of these experimental products at a level high enough to allow us to actually use them!

We are only in the early stages, but we now have an environment in which people who are collaborating with the consortium write and edit hypertext, and save the results back to our server. Amaya, the browser/editor, handles HTML, XML, Cascading Stylesheets, Portable Network Graphics, and a prototype of Scalable Vector Graphics and Math ML. While we have always developed Amaya on the Linux operating system, the Amaya team has adapted it for the Windows NT platform common in business, too. I now road test the latest versions of these tools as soon as I can get them, sending back crash reports on a bad day and occasionally a bottle of champagne on a good one.

We are using our open source Java-based server, Jigsaw, for collaborative work. For example, Jigsaw allows direct editing, saves the various edited versions of a document, and keeps track of what has been changed from one version to the next. I can call

up a list of all versions, with details about who made which changes when, and revert to an older version if necessary. This provides everyone with a feeling of safety, and they are more inclined to share the editing of a piece of work. Jigsaw and Amaya allow our team space to come alive as our common room, internal library, and virtual coffee machine around which staff members who are in France, Massachusetts, Japan, or on an airplane can gather.

Making collaboration work is a challenge. It is also fun, because it involves the most grassroots and collegial side of the Web community. All Web code, since my first release in 1991, has been open source software: Anyone can scoop up the source code—the lines of programming—and edit and rebuild them, for free. The members of the original www-talk mailing routinely picked up new versions of the original Web code library "libwww." This software still exists on the consortium's public server, www.w3.org, maintained for many years by Henrik Nielsen, the cheerful Dane who managed it at CERN and now MIT. Libwww is used as part of Amaya, and the rest of Amaya and Jigsaw are open source in the same way. There are a lot of people who may not be inclined to join working groups and edit specifications, but are happy to join in making a good bit of software better. Those who are inspired to try Amaya or Jigsaw, want to help improve them, develop a product based on them, or pick apart the code and create an altogether better client or server can simply go to the w3.org site and take it from there, whether or not they are members of the consortium.

We create other tools as we need them, and our tool-creation crew is always much in demand. Meeting registration, mailing-list management, and control for our Web site are typical examples. We are looking forward to the time when we will use public key cryptography to authenticate collaborators. Every now and again the new systems go down, and we pay the price for being on the bleeding edge by having to wait till they are fixed.

But we are gaining more of an understanding of what it will take to achieve the dream of collaboration through shared knowledge.

I expect these tools to develop into a common new genre on the Web. Real life is and must be full of all kinds of social constraint—the very processes from which "society" arises. Computers help if we use them to create abstract *social machines* on the Web: processes in which the people do the creative work and the machine does the administration. Many social processes can be better run by machine, because the machine is always available, it is free from bias, and no one likes to administer these kinds of systems anyway. Online voting is one example, and it's already beginning to happen: ADP Investor Communications and First Chicago Trust have services that conduct online proxy voting for corporate shareholder meetings, and more than a thousand companies are using them.

People are already experimenting with new social machines for online peer review, while other tools such as chat rooms developed quite independently and before the Web. MUDDs are social tools derived from multiuser games of Dungeons and Dragons where thousands of people take on roles and interact in a global, online fantasy world. By experimenting with these structures we may find a way to organize new social models that not only scale well, but can be combined to form larger structures.

Almost a decade ago now, I asked Ari Luotonen to spend three days writing a discussion tool for the nascent Web. It was to be like a newsgroup, except that it would capture the logic of an argument. I'd always been frustrated that the essential role of a message in an argument was often lost information. When Ari was done, anywhere on the CERN server that we created a subdirectory called Discussion, a new interactive forum would exist. It allowed people to post questions on a given subject, read, and respond. A person couldn't just "reply." He had to say whether he was agreeing, disagreeing, or asking for clarification of a point.

The idea was that the state of the discussion would be visible to everyone involved.

I would like any serious issue to be on the Web in hypertext. I would like annotation servers to exist where groups could add links (or sticky yellow things) to documents they want to comment on. Annotation servers are a third-party service allowing a group to share each others' coments on documents anywhere else in the Web. The browser gets the original page and then separately checks annotation servers for comments, which are then superimposed on the page. Imagine having servers for comments in different forums, perhaps family, school, and company. Each point and rebuttal is linked, so everyone can see at a glance the direct agreements and contradictions and the supporting evidence for each view, such that anything could be contested by the people involved. If there was some sort of judicial, democratic process for resolving issues, the discussion could be done in a very clear and open fashion, with a computer keeping track of the arguments. Again, the theme is human beings doing the thinking and machines helping it work on a larger scale, but nothing replacing wisdom in the end.

My hope was that the original "Discussion" idea, and future mechanisms that could evolve from it on the new Web, would move us beyond the historical situation of people hurling mud at each other, of peppering their arguments with personal abuse and vitriol, and replace all that with much more of a reasoned, Socratic debate, in which individual ideas, accusations, and pieces of evidence can be questioned or supported.

What Ari and I were trying to do was create a machine that would do the administration for, say, a court, or working group, or parliament. The initial trial was a discussion for the sake of discussion, and it didn't make a big splash. There are now a number of software products for doing some of these things. To actually emulate a courtroom or a democratic voting process, however, the tools need much more development. I long for a

move from argument by repetition of sound bites to a hypertext exposition that can be justified and challenged—one that will allow us to look up and compare, side by side, what politicians, or defendants and accusers, actually say, regardless of what is claimed in television commercials and nightly news interviews.

Because of low overhead, social machines will allow us to do things we just couldn't do before. For example, they will allow us to conduct a national plebiscite whose cost would otherwise be prohibitive. This would, of course, like all the benefits of this new technology, be biased toward those with Internet access. This is just an example to show that we can reassess what is possible; I am not advocating a move from representative democracy to direct democracy. We should be careful not to do things just because they are possible.

Perhaps the Web will enable more organic styles of management, in which groups within a company form in a local, rather ad hoc fashion. They could be made self-forming like a newsgroup, but with constraints that ensure that whoever joins is needed for the work of the company and is covered by sufficient budget. Beyond that, the company doesn't have much conventional structure. When someone has a task to perform, they associate with whomever they need to get it done. People make commitments and negotiate them between groups, without having to go to a manager. The whole organic organization could grow from a seed of a few digitally signed documents on the Web, over the substrate of an electronic constitution that defines how the social machines operate. Provisions for amending the constitution would provide for mutation. A few minimalist rules would ensure fairness.

While there is great excitement because these new social systems are essentially independent of geography, race, and religion, they will of course isolate those in developing countries who cannot afford or have no option to access the Internet. At once the great equalizer and the great divider, the Web highlights—as do

clean water and health care—the necessity of those better off to care for but not simply control those less advantaged. I do no more than touch on that urgent debate here.

The stage is set for an evolutionary growth of new social engines. The ability to create new forms of social process would be given to the world at large, and development would be rapid, just as the openness of Web technology allowed that to bloom.

My colleagues and I have wondered whether we should seed this process using the consortium itself. We could construct the consortium social machine out of the many machines that make up working groups and staff meetings and so on. We could allow a set of working groups that can be shown to form a tight self-reliant cluster to secede and form a new peer "clone" consortium. The rules would have to include more than a newsgroup-like vote; budgets and contributions would have to balance, and responsibility would have to be accepted. In theory, we could then generalize this new social form. Then anyone could start a consortium, when the conditions were right, by pushing a few buttons on the Web page of a virtual "consortium factory."