

Journal of Digital Information, Vol 1, No 1 (1997)

The World Wide Web - Past, Present and Future

Tim Berners-Lee

Tim Berners-Lee was awarded a Distinguished Fellowship of the [British Computer Society](#) on July 17, 1996 at the new British Library in London. The following is a transcript of his presentation.



It is a great honour to be distinguished by such a Fellowship and I should immediately say two things:

- one is that, of course, the Web was developed by a whole lot of people across the Internet, who discovered about it on Internet User Groups and went away with the ideas and started playing and encouraging each other, and developing a little grass-roots community. The Web owes an incredible amount to those across the Internet, and also to the "bosses who didn't say no" who are now all wearing halos across the planet, and who really enabled this sort of thing to grow to the point where they didn't have the option of saying no.
- the other thing I would say is that the ideas existed before putting it together in the World Wide Web, which is basically trivial.

So what's special about it? What I think we are celebrating then is the fact that dreams can come true. So many times it would be nice for things to be this way but they don't come out for one reason or another. The fact that it did work is just so nice; that dreams *can* come true. That's what I've taken away from it, and I hope that it applies to lots of other things in the future.

I'll go back now over a little bit about the [origins](#), a bit about the [present](#), and just a little bit about the [future](#), very much in overview.

The past

The original intent of the Web was that it should be let's start with a definition the 'universe of network accessible information'. The point about it being a universe is that there is one space. The most important thing about the Web is this URL space, this nasty thing which starts with HTTP. The point of a URL is that you can put anything in there, so the power of a hypertext link is that it can point to absolutely anything. That is why, whereas [hypertext](#) had been very exciting beforehand, and there had been a little community that had been happily going on for several years making Hypertext systems that worked across a disc or across the file system, when the Web allowed those hypertext links to point to anything and it suddenly became a critical mass, it became really exciting. Maybe that will happen to some other things as well.

In fact the thing that drove me to do it (which is one of the frequently asked questions I get from the press or whoever) was partly that I needed something [to organise](#) myself. I needed to be able to keep track of things, and nothing out there, none of the computer programs that you could get, the spreadsheets and the databases, would really let you make this random association between absolutely anything and absolutely anything, you are always constrained. For example, if you have a person, they have several properties, and you could link them to a room of their office, and you could link them to a list of documents they have written, but that's it. You can't link them to the car database when you find out what car they own without taking two databases and joining them together and going into a lot of work. So I needed something like that.

I also felt that in an exciting place like CERN, which was a great environment to be in, and to start this. You have so many people coming in with great ideas, doing some work, and [leaving with no trace](#) of what it is they've done and why they did it -- the whole organisation really needed this. It needed some place to be able to cement, to put down its organisational knowledge.

And that idea, of a team being able to work together, rather than by sequence of grabbing somebody at coffee hour and bringing somebody else into the [conversation](#), and having a one time conversation that would be forgotten, and a sequence of messages from one person to another. Being able to work together on a common vision of what it is that we believe that we are doing, and why we think we are doing it, with places to put all the funny little 'this is why on Tuesday we decided not to do that'. I thought that would be really exciting, I thought that would be a really interesting way of running a team, maybe we could work towards that goal: that dream of the 'self-managing team'.

So, that was why these were the original goals. [Universal access](#) means that you put it on the Web and you can access it from anywhere; it doesn't matter what computer system you are running, it's independent of where you are, what platform you are running, or what operating system you've bought and to have this unconstrained topology, which because hypertext is unconstrained it means you can map any existing structures, whether you happen to have trees of information or whatever. As people have found, it is very easy to make a service which will put information onto the Web which has already got some structure to it, which comes from some big database which you don't want to change, because hypertext is flexible, you can map that structure into it.

In the early days, talking over tea with somebody, I was comparing it to a bobsled: There was a time before it was rushing downhill that there was quite a lot of 'pushing' to be done. For the first two years, there was a lot of going around explaining to people why it was a really good idea, and listening to some of the things that people outside the hypertext community come back with.

The hypertext community, of course, knew that hypertext was cool, and why doesn't everybody like it? Why doesn't everybody use it? People felt that [hypertext was too confusing](#) -- the 'oh, we'll be lost in it, won't we?' syndrome. Also I was proposing to use an SGML type syntax. SGML at the time was mainly used in a mode whereby you would write an SGML file and you would put it in for batch processing perhaps overnight on an IBM mainframe and with a bit of luck you would find in the morning a laser printed document. But the idea of doing this SGML parsing and generation of something that could be read in real time was thought to be ridiculous. People also felt that HTML was too complex because 'you have to put all those angle brackets in'. If you're trying to organise information get real you're not going to have people organising it. You 'can't ask somebody to write all those angle brackets just because they want to make something available on an information system, this is much too complex'.

Then there was also a strong feeling, and a very reasonable feeling at CERN, that 'we do high-energy physics here'. If you want some special information technology, somebody has bound to have done that already, why don't you go and find it? So

that took me with a colleague, the first convert, Robert Cailliau, to the [European Conference on Hypertext](#) at Versailles where we did the rounds of trying to persuade them all these people who had great software and great interfaces and had done all the hard bits to do the easy bit and put it all on-line. But having, perhaps due to lack of persuasive power, not succeeded in that, it was a question of going home and taking out the NeXT box.

Using NeXT was, I think, both a good and a bad step. The [NeXT box](#) is a great development environment, and allowed me to write the WorldWideWeb Program. (At that time it was spelled without any spaces. Now there are spaces, but for those of you who are interested in that sort of thing, there are no hyphens). So the WorldWideWeb was a program I wrote at the end of 1990 on the NeXT. It was a browser editor and a full client. You could make links, you could browse around it was a demonstration which was fine but, of course, very few people had NeXT and so very few people saw it.

At CERN, there was a certain amount of raised eyebrows and it was clear that we wanted it on MAC, PC and Unix platforms but there wasn't the manpower to do it. So it we went around conferences and said 'hey, look at this. If you have a student, please suggest they go away and implement this in a flashier way and on one of the platforms please'. There was a couple of years of that.

There was also the Line Mode Browser which was the first real proof of universality. The [Line Mode Browser](#) was a very simple Web browser that runs on a hard-copy terminal. All you need is the ASCII character set, carriage line feed, and you can browse and print a node out, with little numbers by all the links at the bottom and you can choose a number (I mention these things just because sometimes it's worth remembering that the path through A to B is sometimes through C,D and E F and G in totally different places). It was necessary to put the Line Mode Browser out to get other people who didn't have NeXT able to access the Web, so that nobody has the excuse not to be able to access it. The next thing I see is a newspaper article saying that the WorldWideWeb is a system for accessing information 'using numbers'.

There is a [snowball effect](#) here. It is very difficult when you produce a new information system. You go to someone and say 'hey, look in here' and they say 'What? What have you got?' 'It's all about the World Wide Web', they say 'big deal'. So you say 'why don't you put some more information in here' and they say 'who's looking in it?' and you have to say 'well, nobody yet because you haven't put any information in yet'. So you've got to get the snowball going. Now that's happened and you can see the results.

Initially the first thing we put on the Web was the [CERN phone book](#) which was already running on the mainframe. We did a little gateway which made the phone book appear in hypertext with a search facility and so on. For the people at CERN there was a time when WWW was a rather strange phone book program -- with a really weird interface! During that time gopher was expanding at an exponential rate and there was a strong feeling that gopher was much easier, because with gopher you didn't have to write these angle-brackets.

But the Web was taking off with distributed enthusiasm; it was the system administrators who were working through the night when it got to 6 o'clock in the morning and they decided that 'hey, why bother going home' and they started to read [alt.hypertext](#) (yes, hypertext an alternative news group, one of those alternative sciences). Alt.hypertext is where you had to discuss this sort of thing. These systems administrators were the people who would only read the alternative news groups and they would pick up the software and play with it. Then by 8 o'clock in the morning you'd have another Web Server running with some new interesting facet and these things would start to be linked together.

There were some twists and turns along the winding road, there was my attempt to explain to people what a good idea URLs were they were called [UDIs](#) at the time, Universal Document Identifiers and then they were called Universal Resource Identifiers, then they were called Uniform Resource Locators (in an attempt to get it through the IETF, I consented that they could be called whatever they liked). I made the mistake of not explaining much about the Web concepts, so there was a 2-year discussion about what one could use identifiers' names and addresses for. It is pretty good if you're into computer science, you know you can talk for any length of time about that kind of thing without necessarily coming to any conclusion.

It's worth saying that I feel a little embarrassed accepting a fellowship when there are people like Pei Wei a very quiet individual who took up the challenge. He read about the World Wide Web on a newsgroup somewhere and had some interesting software of his own; an interpreted language which could be moved across the net and could talk to a screen. Basically, he had something very like Java, and as he went ahead and wrote something very much like Hot Java, the language was called [Viola](#) and the browser was called 'ViolaWWW'. It didn't take off very quickly because you had to first install 'Viola' nobody understood why you should install an interpreter, and then this 'WWW' in a Viola library area. You had to be system administrator to do all that stuff, it wasn't obvious. But in fact what he did was really ahead of his time. He actually had Applets running. He had World Wide Web pages with little things doing somersaults and what have you.

Then there was a serious turning point when someone at NCSA brought up a copy of 'Viola' and Mark Andreessen and company saw it and thought 'Hm, we can do that'. Mark Andreessen worked the next 14 nights, or something, and had [Mosaic](#). One other thing he did was put in images, and after that the rest is more or less history. Nothing had really changed from the Line Mode Browser in that Viola was just a browser, it was not an editor. And the same for Erwise which had preceded it. In fact there is another one called Cello which had been written for the PC which preceded Mosaic, and in each case they wrote a World Wide Web client which is a piece of software which can browse around the Web, but unlike the original program you couldn't actually edit or make links very easily.

I think this was partly because NeXTStep software has got some neat software for making editable text, which is difficult to do a WYSIWYG word processor from the ground up. But it is also because when you get to the Browser, you get all excited about it, you get people mailing you and end up having to support it and answer questions about it. Mark Andreessen found himself deluged by excitement in Mosaic, and still we didn't have anything which could allow people to really write/create links easily with a couple of key strokes, until NaviPress -- who's heard of NaviPress? -- a little Company bought by AOL and now called AOL Press. They are still there, and a number of other editors which actually allow you to go around and make links although still not as intuitively as I would have liked.

So those are some of the steps, there are lots of other ones and many anecdotes, but this was the result as seen from CERN [refers to Figure showing straight line growth of use of WWW on CERN server, with vertical axis on a logarithmic scale]. This shows the load on the first WWW server. By current terms it's [not a very big hit rate](#). Across the bottom is from July '91 to July '94 and there is a logarithmic scale up the side of total hits per day. The crosses are week days and the circles are weekends and you can see what happened -- I call that a straight line -- you can see that every month, when I looked at the log file it was 10 times the length of the log file for the same month the previous year. There are a couple of dips in August and there are a couple of places where we lost the log information when the server crashed and things.

People say 'When did you realise that the Web was going to explode like this?' and 'when did it explode?'. In fact if you look, there was the time when the [geek community](#) realised that this was interesting, and then there was the time when the more established computer science and high energy [physics community](#) realised that this was interesting, and then there is when [Time and Newsweek](#) realised it was interesting. If you put this on a linear scale, you can pick your scale and look for a date on

which you can say it exploded, but in fact there wasn't one. It was a slow bang and it is still going on. It's at the bottom of an 'S' Curve and we are not sure where the top is.

The present

And then after the bang we are left with the post-conceptions (the reverse of pre-conceptions). One of those was that because the first server served up Unix files, there was an assumption that those things after the HTTP: had to be Unix file names. A lot of people felt locked into that, and it was only when Steve Putts put up an interesting server where URLs were really strange but would generate a map of anywhere on the planet to any scale you wanted and with little links to take you to different places and change the scale.

After a few other really interesting servers which had a different sort of information space, the message got through that this is an opaque string and you could do with it what you like. This is a real flexibility point, and it's still the battle to be fought. People try to put into the protocols that a semi-colon here in the URL will have a certain significance, and there was a big battle with the people who wrote 'Browser' that looked at the '.html' and concluded things about what was inside it wrong! URL is not a file name, it is an opaque string and I hope it will represent all kinds of things.

People kept complaining about URLs changing - well, that was a strange one because URLs don't change, people change them. The reasons people change them are very complex and social (and that gets you back into the whole naming and addressing loop) but there was a feeling for a while that there should be a very simple, quick cure to making a name space, in which you would just be able to name a document and anybody would be able to find it.

After a lot of discussions in the ITF and various fora, it became clear that there was a lot of social questions here, about exactly who would maintain that, and what the significance of it was, and how you would scale it. In fact there is no free lunch, and it is basically impossible.

There was the assumption that, because links were transmitted within the HTML that they had to be stored within HTML files, until people demonstrated that you could generate them, on the fly, from interesting programs. And from the assumption that clients must be browsers it seemed to follow that they can't be editors - for some reason although everybody has got used to WYSIWYG in every other field, they would not put up with WYSIWYG in the Web.

But people had to write HTML -- you have to write all those angle brackets. It was one of the greatest surprises to me that the community of people putting information on line was prepared to go and write those angle brackets. It still blows my mind away, I'm not prepared to do it, it drives me crazy. But now we hear back from these people who got so into writing the angle brackets, that HTML is far too simple; we need so many exciting new things to put in it because we need to be able to put footnotes, and frames, and diagonal flashing text that rotates and things. Didn't things change over those few years?

And where are we now? Well, what you actually see when you look at the Web is pretty much a corporate broadcast medium. The largest use of the Web is the Corporation making a broadcast message to the consumer. I'd imagined initially that there would be other uses and I talked a bit about group work business, but clearly once you've put something up, if there is any incentive -- whether it is psychological or monetary or whatever, because your audience is very large, it is very easy for you to push it up the scale, it pays you very much to go for that global audience. You can afford to put in so much more effort if you have got a global audience for your advertising, for your message, subtle or not. So that is what is seen. And there is some cool stuff.

There is VRML; 3-D sites where you wander through 3-Dimensional space, maybe that will become really interesting (actually I think it will happen because to do 3-D on a machine you need a fast processor but you don't need a fast phone line and I think the fast processors are coming a lot faster than the fast phone lines). So 3-D is something which may happen a long time before video.

There are style sheets coming out which will allow you to do that flashing orange diagonal rotating text. You can redo all your company wide documents with the flick of a button just by changing the style sheet without having to change all that HTML.

There's Java, which is really exciting. At last the Web has given the World an excuse to write in a decent programming language instead of C or Fortran. Begging your pardon, there have been object oriented programming languages before now, but if a real programmer programmed in one, typically the boss would come round and say 'sorry, that's fine, son, but we don't program like that in this organisation', and you have to go away and re-write it all in C. Just the fact that the Web has been there to enable a new language to become acceptable is something.

What's the situation with the Web itself as an information space? From the time when there was more than one browser, there was tension over fragmentation. Whenever one browser had a feature, an adaptation of the protocol, and the other one didn't, there was a possibility that the other one would adapt, would create that feature but use a slightly different syntax, or a very different syntax, or a deliberately different syntax. You get places where you find a little message which says, 'this page has been written to work with Mosaic 5.6 or NetScape 3.0 or Internet Explorer 2.8 or whatever it is, and it's best for you to use that browser'.

And now? Do you remember what happened before the Web? Do you remember this business when you wanted to get some information from another computer: you had to go and ask somebody how to use this telnet program, you had to take it to someplace and ftp files back onto a floppy disk, you picked the floppy disk up and went down the corridor, and it wouldn't even fit in your computer!

When you got yourself a disk-drive that would take it, then the disc format was wrong, so you got yourself some software, and with someone's help you could read the format on it, and what you got was a nice binary WordStar document and there was no way you could get it into Word Perfect or Word Plus 2.3 - remember that? Do you remember how much time you spent doing all that?

Well, the people who put these little things at the bottom of their web pages saying this is best viewed using 'Foobar' browser are yearning, yearning to get back to exactly that situation. You'll have 17 Web browsers on your page and you'll get to little places which say 'now please switch to this' and 'now please switch to that' and suddenly there is not one World Wide Web, but there a whole lot of World Wide Webs.

So if any of you have got Web Masters out there put those little buttons on there saying this is best used using a particular browser, suggest they put 'this is best used using a browser which works to the following specifications: HTML 3.2', or something like that. You can go back this evening; email them from your homes, and tell them that I just mentioned it.

So there is a tension of fragmentation, what are we going to do about it? In 1992 people came into my office, un-announced, from large companies, sometimes more than one company at a time. I remember one in particular when four people came and sat down around a table and banged it and said 'Hey, this Web is very nice but do you realise that we are orienting our entire business model around this. We are re-orienting the company completely, putting enormous numbers of dollars into

this, and we understand the specifications are sitting on a disk you have somewhere here. Now what's the story, how do we know it is still going to be there in 10 years, and how do we put our input into it?

I asked, of course, what they felt would be a good solution to that, and I did a certain amount of touring around and speaking to various institutes, and the result was I felt there was a very strong push for a neutral body. Somewhere where all the technology providers, the content providers, and the users can come together and talk about what they want; where there would be some facilitation to arrive at a common specification for doing things. Otherwise we would be back to the Tower of Babel.

So hence the World Wide Web Consortium. The Consortium has 2 hosts; INRIA in France for Europe, and MIT for North America. We are also looking at setting up various things in the Far East. We have 145 members at the last count (maybe it's 150 now -- it seems that the differential between my counting and my talking about it is 5). We are a neutral forum - we facilitate, we let people come together. We actually have people on the staff who have been editing Web specs, are aware of the architecture, are basically very good. They can sit in on a meeting and edit a document, know when people are saying silly things, and produce a certain amount of advice. We have to move fast.

We are not a standards organisation, I'm sorry. We do not have meetings from every one of our 150 or whatever it is, countries in the world sitting round, and we do not have 6 month timescales. Sometimes we have to move extremely rapidly when there is a need for something in the marketplace and the community wants to have a common way of doing it. So we don't call what we do 'standards' we call them 'specifications'.

We have just introduced a new policy by which we can simply ask the members whether they think something is a good idea, and if they do then we call it a 'recommendation' as opposed to a 'standard'. In fact what happens is that when we get together, the engineers who know what they are talking about from the major players (they are primary experts in the field), write a little piece of specification, put their names on it, and it's all over bar the shouting. Everybody takes that 'spec' and runs with it, de-facto 'standards' arrive in most cases.

But every area is different and so we have to be very flexible. Some areas we have to consult, we have to be more open, there are more people who want to be involved. In some areas we have to just move extremely rapidly because of political pressure.

At the same time we like to keep an eye on the long-term goals, because although the pressures are fairly short-term there is a long-term architecture. There are some rules in the World Wide Web; like the fact that URLs are opaque; like the fact that you don't have to have HTTP at the beginning of a URL but you can move onto something else; like the fact that HTTP and URLs are independent specifications and HTML is independent of HTTP, you can use it to transport all kinds of things.

If, originally, the specs had fixed that the World Wide Web uses HTTP and HTML we wouldn't have Java applications or other things being transported across the Web. We wouldn't be able to think about new protocols.

The future

It's worth saying a word about the long-term goals. There is still a lot of work before this can be an industrial strength system, so that when you click on the link you know you are going to get something.

There are a lot of things that have got to change, such as **redundancy** which has got to be able to happen, just fixing everything 'under the hood' so that you can just forget about the infrastructure. Something which is very complicated, involves some pretty difficult problems in computer science, and it's important.

I have a horizontal scale between the individual human interaction at the end, through to the corporation talking to the masses. I'd originally imagined that the point about the Web was that you would also be able to have **personal diaries**, and in that personal diary you'd be able to make a note, and you'd be able to put a pointer to the family photograph album, and your brother's photograph album, which are just accessible to the family, or the extended family.

You would be able to put a pointer to a meeting you've got to go to at work, but the **meeting agenda** would be just visible to and used by a little group of people working together, and that in turn would be linked to things in the organisation of the town you are living in, such as the school. Imagine that you have a range of things going up through what is called the Intranet (the World Wide Web scaled down for corporate use), to the whole global media thing, and that this would all be one smooth continuum.

I thought it was simple, we just needed to get browser/editors which were good and then we would be able to play. To a certain extent that's true. When we do have browser/editors we'll be able to do a lot more, but there is a lot more that you need. You need to **have trust**; you need to be able to make sure that other people don't see those photograph albums and what have you. There is a lot of infrastructure that has still to be put together, but I am very interested in the Web being used across that scope.

I'm also interested in these machines that we all have on our desks being actually used to help us. What they are doing at the moment is delivering things for us to read, decisions for us to make and information for us to process. *For us to process!* Hey, what about these computer things? I thought the idea was *they* were supposed to **do some of the work**. At the moment they can't. They could, in fact, do it when it's a database but they haven't a chance on the Web because everything on the Web is written in bright shining pink and green for your average human reader, who can read English (who can read pretty bad English at times), so if you and I have difficulty parsing it, going out and asking a machine to solve the problem is pretty difficult at the moment.

Let's suppose there is a house for sale and you want to buy it. You would like to know that person really owns it. Suppose you don't have a Land Registry, so you go and you find the Title Deeds, which are on the Web, as are all the transfers of ownership going way back. They are there, but it's a lot of work to go back through all of them unless they are put in a form that is actually a semantic statement, some knowledge representation language statement.

Knowledge representation is another thing that people have played with, but it really hasn't taken off in a tremendous way on a local scale. Maybe it is something that, if we can get the architecture right globally, then that would take off too. Then you would be able to simply ask your computer to go out and find an interesting house, the sort of house you like, within your price range, and see if it is really owned by the person who is selling it (or whether in fact they sold off half of the back garden 10 years ago but they hadn't told you). It would be able to go and make all the assumptions, it would be able to figure out in fact whether the documents it reads it ought to believe, by tracing through the digital signatures and what have you.

Those are some long-term goals. They are not things that the press, the consortium, the members, the newsgroups, talk about all the time, but they are things we are trying to keep in the back of our minds.

I'll go through very rapidly the areas that W3C is actually developing or could develop. There are basically 3 areas of work:

The **User Interface** and **Data Formats**, the parts of the architecture and protocol which are affected by, and specifically affect, the sort of society that we can build on the Web. The sort of things that are in the user interface area are the continual enhancement of HTML for more and more features, putting different sorts of SGML documents onto the Web, solving the internationalisation problem (or at least trying to take those pieces of the internationalisation problem, type-setting conventions, such as type-setting in different directions and character sets) and trying to take those solutions which exist and show how you can use them in a consistent way on the Web. Style sheets, graphics in 3-Dimensions. The PNG format, for example, is a new graphics format to replace the graphics interchange format because it's bigger and better, which we have been encouraging (although not doing ourselves). Most of the user interface and data formats work is done in Europe.

There is the whole area of **Web protocols** in Society, security and payment and the question of how parents can prevent their children from seeing material which they don't want them to be viewing until they are old enough. It is this pressure to protect a child, until the age of digital majority, particularly in the United States but also in Germany and various other countries, that has produced the Platform for Internet Content Selection, or PICS system. This is an initiative which has produced specifications which should, I hope, be in software and usable by the end of 1996.

There are other exciting things on the horizon, such as protocols to actually transfer semantic information about intellectual property rights. Can you imagine taking the licence information on the back of a floppy disk, one of those in such small type that if you blew it up to a readable size it would probably be poster size, and actually trying to code that up into some sort of semantic language I can't, but maybe we can work in that direction. Questions of how to find the demographics of who is looking at your site without infringing the privacy of any individual person.

The third area of Web architecture is looking at the **efficiency and integrity** of the Web. How do you prevent the problems of dangling links; find out when you have linked to a document which no longer exists; rapidly and painlessly? How do you get copies of heavily used documents out to as many places as you can, all over the planet, and having done that, how does the person in the arbitrary place find out where the nearest one is? These are part of the unsolvable naming problem.

In general, we are aiming to bring the thing up to industrial strength. We had a workshop about this recently. There is the question of whether these things we find on the Web are really objects and what does that mean? Does this mean that the distributed object world should somehow merge, there should be some mapping between Web objects and distributed objects. What does this mean?

And that raises the question of **mobile code objects**, which actually move the classes around. There are lots of exciting things going on, not that the average user would notice apart from the fact that they get little gismos turning corners of the tops of their Web pages when Java applications come over.

There is just one more thing that I want to emphasise. I initially talked about the Web, and said that I wanted it to be **interactive**. I meant this business about everybody playing at the level where you have more than one person involved but not the whole Universe. Perhaps you've got a protected space where you can play.

I feel that people ought to be able to make annotations, make links and so get to the point where they are really sharing their knowledge. I talked about interactivity. I found people coming back to me and saying 'Isn't it great that the Web is interactive' and I'd say 'Huh?'. 'Well you know you can click on these buttons on forms and it sends data right straight into the machine'.

I felt that if that is what people meant by interactivity then maybe we need another word (I say this with total apology because I think people who make up new words are horrible) but let's just for the purpose of this slide talk about intercreativity, something where people are building things together, not just interacting with the computer, you are interacting with people and being part of a whole milieu, a mass which is bound together by information.

Hopefully with the computers playing a part in that too. To do that we need to integrate people with **real-time video** that you hear so much about. Why isn't it better integrated with the Web? Why can't I when I go to the library the virtual library that is find people's faces and actually start talking to them? Why don't I meet somebody in the library?

The nice thing about the virtual library is that you are allowed to talk in it, except that talking protocols haven't been hooked into the Web protocols yet, so we just need to do a little hooking together (Ha! 'a little bit of hooking together there' sounds like 3 years work of solid standardisation meetings).

How about having objects that you can manipulate. I'd like to be able to hold a virtual meeting in a 3-dimensional area where there is a table and where you can move the chairs around, and when I move the chairs you see it happen.

We could build graphs and models, mathematical models, real models, engineering models, little models of new libraries, to see if we can make them look nice sitting next to St. Pancras Station or something. I'd like to be able to see all that happen in the Web so that means building into the infrastructure objects; objects which know how to be interacted by many people at once and being able to update their various many instances and copies throughout the World.

The military folks use 3-D digital simulation technology for playing tank battles and maybe there will be some good stuff coming out of that, I don't know. But a very simple thing would be to notify somebody that something has changed. It's great having this model of global information - you write something, I go in and I change it, and put an important little yellow post-it sticker on it, but if you don't find out that I've done it then it's not very much use, so we need to have ways of notifying both people and machines that things have changed.

I would like to see people get more involved in this; at the moment, it doesn't look like one great big television channel, but lots and lots and lots and lots of very shallow television channels and basically the mouse is just a big television clicker. There must be more to life.

So, let me conclude with a few challenges that as a community we have. One is making the most of this **flexibility**, we have got to, we need to keep it flexible. We need to be able to think our way past the Web as a set of linked hypertext documents.

Hopefully pretty soon the Web infrastructure, the information space, will be just a given, like we assume IP now (we don't worry about IP, although we should, because it's running out of address space and all kinds of stuff and nobody is funding the transatlantic links). We just kind of assume that Internet Protocol is there, the Internet is there and we are worrying about how we build the Web on top of it. We've got to make sure that there is somebody there having the next bright idea and can use that flexibility to make something which has got a totally different topology which is used to solve a totally different problem. To do that, we have got to make sure that we are not, in our designs, **constraining the future evolution**, we're not putting in those silly little links between specifications.

Let me give you just one example: It is possible with some browsers to put a piece of HTML on the Web. The server delivers it to the browser and inside one of the tags is an attribute, and the attribute value is quoted, and inside the attribute value is a quoted string. It's normally used to be able to write 10 for something like a point size or a width or a whatever but now you can put a little piece of Javascript in there and some browsers, if they don't see 10 but something in curly brackets, they will

just send it off to the Javascript interpreter.

Now if you've actually got a Javascript interpreter this is dead easy. You can do that in 2 lines of code; just take the curly brackets off and call Javascript, but just think what's happened. In ten years time, to figure out what it meant, not only do you have to look up the old historical HTML space, but you have also got to find Javascript. Javascript is going to be changing and so you thought you had a nice, well defined language, but it's just one line's reference from that specification to the other specification.

In fact, you've got a whole big language specification except that in one part of it it's got angle brackets, and the other part of it has curly brackets and semi-colons; and they are totally different, one thing is totally incomplete and the other is self modifying. And so, by not saying 'by the way this document is in HTML and Javascript' 2.0 -- that one little trap -- then it's the sort of thing which could trip us up later.

The third thing which is really important is that we have to realise that when we define these protocols and the data formats, we are defining things like the topology of the information. We are defining things like who can get access to what information. We are defining things about privacy; about identity; how many identities you can have; whether it is possible to be anonymous; whether it is possible for some central body to do anything at all; whether it is possible for a central body to do lots of things like find out the identity of anonymous people.

Whether there is a right for two people to have a private conversation, which we rather assume at the moment because they can go into the middle of a big field, but does that right hold in Cyberspace? If it does, does this mean that the world will fall apart because terrorism will be so easy? Do all these questions about society come back to the protocols we define, which define the topology in the properties of Cyberspace.

So if you think you're a computer programmer, if you think you're a language designer, if you think you're a techie; and one of the nice things about being a technie is that you can forget all that ethics stuff because everybody else is doing that -- and thank goodness you didn't have to take those courses -- you are wrong.

Because when you design those protocols you are designing the space in which society will evolve. You are designing, constraining the society which can exist for the next 10-20 years.

I'll leave you with that thought.

Tim Berners-Lee
17 July 1996