

Wire: Email Interview with Chris Lilley

[Chris Lilley](#) submits to an interview by email.

I represent [JISC](#) at the Advisory Council meetings of the [World Wide Web Consortium \(W3C\)](#). Most of the delegates are representing commercial companies, whereas I am effectively representing the UK Higher Education sector! W3C member companies are given advance information in confidence, and I am currently working with W3C to see how I can involve UK HE in the work of W3C without violating that confidence. This position is funded through the [Advisory Group on Computer Graphics \(AGOCG\)](#) and covers 25% of my time. I run a mailing list (agocg-w3c@mailbase.ac.uk) where I notify people of the [latest W3C work](#) and the implications to UK HE is discussed. My aim is to ensure that the academic sector has a voice to influence the development of the Web.

HTML and Stylesheets is an area I have been involved with for some time; I am an active member of the [IETF HTML Working Group which standardised HTML 2.0](#); I am a contributor to the [www-talk](#), [www-html](#), [www-style](#) and [www-font mailing lists](#) and have helped with the development of the [Cascading Style Sheet specification](#).

Working in a [Computer Graphics Unit](#), the issue of Graphics on the Web is of great interest to me, and I gave a paper on the subject of "[Quality Graphics for the Web](#)" at [WWW4](#). I am one of the [authors of the specification](#) for [Portable Network Graphics \(PNG\)](#).

... and how did you get into this area?



Chris Lilley in a frog t-shirt

In late 1992 I got hold of the CERN linemode browser. After poking around the CERN web site for a while, I thought "this is much the same thing as Gopher" and forgot about it. (I never saw the NeXT client, or Viola, until later on). In spring 1993 the first beta of Mosaic for X was released which was very different - a full Motif windowing application with inline images. At the time, I was writing a [series of teaching materials on Computer Graphics](#) and I saw an immediate teaching use which was presented at an international workshop on Graphics and Visualisation Education (Barcelona, 1993). The CGU web went live in August 1993. I later went to the seminal First International Web Conference at CERN, where I participated in the 'Education and the Web' workshop. You can read [my position paper](#). After that it just sort of snowballed.

Buzz

[data](#), [mobile](#), [software](#), [java](#), [html](#), [css](#), [browser](#), [graphics](#), [sgml](#), [dtd](#), [windows](#), [multimedia](#), [linux](#), [png](#), [rfc](#), [visualisation](#), [rtf](#), [gopher](#), [interoperability](#), [standards](#)

Citation

[BibTex](#)
[RIS](#)

How significant an impact do you think that mobile code, such as Java applets, will have on the development of the Web as an information resource?

Mobile code in general is clearly a very significant development but will require some work on object models, interoperability and security. Java solves most of the security problems and I am pleased by Sun's open attitude with Java.

Plug-ins are a sort of mobile code, and people who use MS-Windows are getting all excited by Netscape plug-ins at the moment. The Unix version of Netscape does not support plug-ins, however. The Mac version does, I gather, but of course the MS-Windows plug-ins will not work. When you compare a mobile code technology that is platform, browser, and even browser-version specific (like Netscape plug-ins) with a technology that is platform and browser independent (like Java, though there are others) it is hard to see what the big deal is.

Any developer given the choice between

- a) produce one Java applet that runs on everything that is Java-enabled
 - b) produce the Netscape 2.0 Win 95 plug-in and the Netscape 2.0 Mac plug-in and the Microsoft Internet Explorer plug-in for Win 95 and the one for NT and the plug in for...
- ... is surely going to pick option a. Academic developers simply do not have the time to do all these multiple versions, even if they do have the multiple development platforms.

...and how significant is VRML in the same context?

A 3D non-immersive virtual hyperworld could be very useful for constructing an information resource, because people can use real-world orientation skills for navigation. This has already been shown with text-based virtual environments in an academic information context, for example with BioMOO.

The problem is that VRML 1.0 has nothing to do with VR. It is (yet another) static 3D scene description language. It has inline images and links, but no behaviour can be specified. There is no real-time communication with the server. If some-one else is viewing the same VRML 1.0 file that you are, you don't see them - far less interact with them.

There was talk at WWW4 about adding behaviour and interaction to VRML 1.1 and 2.0. That will be the hard bit. VRML 1.0 was essentially just a question of asking Silicon Graphics about using their existing Open Inventor format, so that was quite easy.

So the short answer is, there is lots of potential and scope for experimentation and development, but I don't know about timescales for really using VRML for serious applications.

A friend buys a PC, relatively fast modem and an internet connection. (S)he asks you what browser they should either buy or obtain. What do you advise?

Get several and see which works best - most are free, anyway. More specific recommendations - well assuming the PC to be a dual-boot Linux/ MS-Windows machine running X-Windows ;-) I would say use the Linux version of Netscape 2.0 as a working browser. Unix versions can use gzip'ed files which are much smaller, so downloading PostScript or Acrobat files over a modem becomes feasible. I would also suggest the Linux version of Arena beta-1e for experimenting.

Alternatively, if restricted to MS-Windows 95 I would suggest Microsoft Internet Explorer 2.0 which seems to display faster than Netscape, particularly for forms, and seems to be the "browser du jour" on that platform.

If this friend was not a native English speaker, I would recommend the Alis browser, which implements the Internationalisation extensions so that Arabic, Hebrew, Cyrillic etc documents can be displayed and also so that documents in the language of choice can be requested. I see that Netscape 2.0N (the X version, at least) can also request documents in a specific language.

Many people have recently commented on the relationship between SGML and HTML. Some speculate that they will one day merge into a "unified" language; some others see HTML diverging into various esoteric directions. What do you think?

Well, HTML is implemented in SGML so that is rather like asking if, say, a wordprocessor (written in C) and the C language will one day merge.

People have been talking about putting other types of SGML document on the web, that use DTDs that are not related to HTML. This has happened, and there are browsers such as Panorama which will read such files. The general SGML publishing community has a rather different focus from the Web community. For example, it is considered appropriate to abandon processing SGML documents on detecting any sort of error, while with the HTML DTD there is an application convention that says do as much error recovery as possible.

Regarding HTML, there is active work by W3C, IETF and others to capture the lessons of the last two years and not only produce updated HTML DTDs that describe newer things - tables, embedded applets, internationalisation, and style sheets - but also to make change easier in the future. For example, modular DTDs that are easier to change; marked sections in HTML, and graceful degradation.

Is there any hope for Gopher, or will the Web and other resources make it (eventually) obsolete?

Tim Berners-Lee defined the Web as "the universe of network accessible information". It is not restricted to just http. Thus, Gopher is part of the Web. There is existing legacy data on Gopher which is still valuable. New projects would however be ill-advised, in my view, to select Gopher over HTTP as a means of delivery.

Electronic Commerce; significant numbers of Web users, especially in the US, buy goods and services through the Web. With the current state of data protection/encryption, would you feel comfortable in sending your credit card number to a company or service through your Web browser?

No. Then again, people read it out over the phone to mail order companies and hand their cards to shop assistants, which is not secure either.

Companies like First Virtual and DigiCash provide alternatives to sending a credit card number over the Web.

One of the most frequent complaints from UK Web users is the slow speed of accessing some Web sites, especially those held in the US during times of high network traffic. What, if anything, should be done (and by whom) to alleviate this problem?

The bandwidth within the UK is extremely good, and we lead Europe and large parts of the US in that area. This means it is just about adequate to the task for accessing academic UK sites. Commercial sites, of course, are often on slow links.

The problem lies in the congested link to the US (and the link to mainland Europe) which is grossly inadequate. The problem is that ten times the bandwidth costs ten times the money, according to the suppliers.

Given that we need more like 50 or 100 times the bandwidth, going to another supplier looks like a useful option. Certainly the current bandwidth must be increased, by whatever mechanism.

Another approach (as well as increasing the bandwidth, I hasten to add, not an alternative to increasing it) is to make more efficient use of the network by sending compressed headers, keeping connections alive, multiplexing connections and so on. There were some experiments a year or two ago using a new protocol, HTTP-NG, which did this. The suggestion was that a pair of proxy servers, one in the UK and one in the US, would talk HTTP-NG to each other over the transatlantic link but would talk ordinary HTTP to other caches and to individual browsers. I have not heard much about this plan lately.

The W3C has sometimes been criticised for the time taken to produce a "finished" HTML3 specification. Is this criticism justified?

No. To explain that, I need to cover a little history.

HTML 3.0 emerged from earlier design work called HTML+, which was presented in Spring 1994 at WWW1 in Geneva. This was largely the work of one individual, Dave Raggett, who then worked for Hewlett-Packard.

HTML 2.0 was started by Dan Connolly in summer 1994 to capture current practice in HTML usage. This work was carried out in the IETF HTML Working Group who took until September 1995 to release the HTML 2.0 specification as RFC 1866, a Proposed Internet Standard. It was widely recognised that this took way too long. The IETF is entirely a volunteer organisation, and anyone can join in.

The IETF also voted to not progress the HTML 3.0 draft, which had been stable since March 1995. At one point this draft was listed as deprecated! However, portions of that draft (eg tables, which had been implemented in several browsers by this time) could be carried forward as the basis for discussion. In my view that was a bad move which caused a loss of confidence in the industry. The HTML 3.0 draft was certainly not perfect and some parts had little implementation experience, but it had been around for a while, people were writing documents in it, and adopting it at that point would have "caught up".

So far, W3C has not figured in this tale. They stated that work on individual parts of a "post 2.0" specification would henceforth be carried out by W3C and its member companies, so that mature drafts could be circulated to the IETF for comment. W3C stated that this would make things quicker. They started to issue a series of [draft Technical Reports](#) containing their work.

Since this happened, the tables draft has been produced, put through the IETF, and is in last call. The most notable example of the W3C helping to speed things up was the issue of inserting multimedia into HTML. Sun, Netscape, Microsoft, and others had all begun to do this, in mutually incompatible ways (app, applet, embed, bgsound, marquee, and so on). The W3C called a meeting of the relevant companies, who were all by this time members of W3C, and very shortly there was a [specification issued](#) which they could all agree to and had helped create.

So, in summary: yes the production of stable documents that people can write pages to has been slow, far too slow, but since W3C has been involved the process seems to have become much quicker.

Web pages can be generally created by either (a) using software to convert from some other format e.g. RTF, to HTML (b) using a package such as HoTMetaL to automate/validate the construction or (c) hand, in a text editor and then validated by e.g. the Halsoft validation mechanism (mirrored at Hensa). Which method do you use, and why?

I distinguish two situations - re-using existing content, and creating a document from scratch. For the former, re-typing is generally out of the question and a conversion tool is required. I use FrameMaker for constructing printed documents, and the converter I use is WebMaker which was originally from CERN and is now a commercial product from Harlequin in the UK.

To create new content, I must admit I just launch into a text editor and start typing. Then again I know HTML fairly well. I always paste in the stuff from another document because I can never remember it ;-) and then validate using html-check which does a formal sgml validation against whichever DTD (HTML 2.0, HTML 2.1, draft HTML 3.0beta, W3C's HTML 1996) I have used. This is not an "over the web" validation, it is a tool that runs locally.

I find this easier than typing into a wordprocessor and converting, because the converter never quite does everything you want and the work involved is often more than just typing by hand.

I have used some structured editors - asWedit is a good one (for Unix platforms with X) - when doing complex tables or playing with less used features like FIG. asWedit has an icon bar and currently invalid elements are greyed out which is a nice way of indicating "you can't put that tag there".

I am currently beta-testing an editor called Symposia which was demonstrated at WWW4. The details of the user interface require some work, but the presentation is excellent - full WYSIWIG editing including tables, forms and so on. This way of working is obviously going to become increasingly important.

What would you like to see happen in the field of the World Wide Web in 1996?

An end to messages that say "you must have browser XYZ version n to view these pages", "set your browser this wide", and pages that come up unusable or even totally blank for people not using the same platform and browser as the document author.

 Facebook

 Twitter

 Google+

 E-Mail

 Pinterest

 LinkedIn

Ariadne is published by [Loughborough University Library](#)

© Ariadne ISSN: 1361-3200. See our explanations of [Access Terms and Copyright](#) and [Privacy Statement](#).