Display and Interactive Languages for the Internet: HTML, PDF, and Java*

Dirk Eddelbüttel
GREQAM-EHESS
Marseilles, France
edd@ehess.cnrs-mrs.fr

William L. Goffe
Dept. of Economics and International Business
University of Southern Mississippi
Hattiesburg, MS 39406
Bill.Goffe@usm.edu

1 Introduction

The rapid rise to prominence of the Internet has left many a bit surprised and even perhaps a bit confused. In particular, there are many tools for displaying and interacting, and it is not immediately clear how they are used, or what is the appropriate tool in which circumstance. This paper describes three tools: HTML (hypertext markup language), PDF (portable document format), and Java. The first two are used to display information; HTML is the standard display technology for the Internet—"web pages" are written in it. However, it suffers from a key deficiency in the eyes of many—it does not allow the author precise control over the layout of information. PDF, on the other hand, does allow precise control. Thus, they fill different niches. Java is a full-featured object-oriented programming language, that is also particularly well suited for interacting with remote computers over the Internet's World Wide Web.

2 HTML

HTML (hypertext markup language) is a key building block of the World Wide Web section of the Internet. When an Internet user, with client software such as the Netscape

*We would like to thank, without implicating, Paul Robbins, James T. Lindley and Jörg Heitkötter for helpful comments.

Navigator browser, retrieves information from a remote web server[1], the information received is encoded in HTML.[2] In short, web documents, from as mundane as a student's home page, to the on-line New York Times, are written in HTML. In addition, HTML links many different computer systems, so it is a "lingua franca" between different systems. This "markup" language tells the browser about both the document's structure and information type: paragraphs, lists, images, types of fonts, and links to other information (thus the term "hypertext," though hypermedia would be more accurate). The browser takes the HTML and then displays the information for the user. HTML even helps call another program to present information if the browser cannot, such as sound, movies, Adobe Acrobat PDF files, or Java[3]. Since HTML displays most information, or provides a link to the appropriate program if it cannot, HTML is a glue that holds the web together.

Thus, for an economist interested in making information available through the Internet, HTML is an essential tool. Fortunately, HTML is easy to learn, and there are many tools for writing it. Some of these tools largely or entirely eliminate the need to learn HTML altogether[4]. However, HTML is not the best choice for presenting working papers on the Internet. As described below, HTML describes content, and not the layout or format of information, so it is difficult to use for technical papers that require precise layouts. In addition, HTML has very limited tools for displaying mathematics. For working papers, the topic of the next section, Adobe Acrobat, is the tool of choice.

It is useful to take a quick look at the history of HTML and the web to understand its current active, but rather chaotic state. Tim Berners-Lee, a computer scientist at CERN (the high energy physics laboratory outside Geneva) introduced HTML and HTTP[5] in 1991 to display and link information for physicists. However, the web was not widely used outside of a closed circle of physicists until it was teamed with Mosaic, the first graphical browser, developed at the National Center for Supercomputer Applications (NCSA) at the University of Illinois. Together, they simplified the use of the Internet, and are an important reason for the Internet's popularity. Part of the team that developed Mosaic left for Netscape; most significant is Marc Andressen, now senior vice-president of technology. Netscape Navigator is currently the leading browser with a commanding market share. Tim Berners-Lee is now director of the World Wide Web Consortium (W3C), which provides leadership for web development[6]. Finally, Microsoft has started to play a significant role in HTML development following their realization that the Internet is essential to their business. Their browser, called Internet Explorer, is said to have caught up with the capabilities of the Netscape browser and reportedly gaining market share.

This fractured leadership, along with vendors adding features to HTML to gain competitive advantage, makes succinctly describing the current state of HTML difficult. The last formally standardized version of HTML was 2.0 of 1995 (most of the work was from 1994 or before), which by today's standard is quite basic[7]. Its major features include various ways of displaying text, such as lists, line breaks, different types and fonts for text, and headers for information; integrated images; links to other information; and forms for entering information on the remote server. Much of HTML 2.0 is described later in an example.

Since then, vendors, particularly Netscape, have added many features to HTML 2.0.

These extensions, along with their superior browser, have lead Netscape to dominance in the browser market. Netscape's extensions to HTML are the most commonly used, and generally have been adopted by other browsers, including Microsoft's Internet Explorer. Thus, HTML 2.0 with Netscape extensions is close to a de jure standard. One effort at further formal standardization, HTML 3.0, failed in 1995. W3C has lead another attempt at standardization, HTML 3.2, which includes many Netscape extensions[8]. It is likely to be accepted as Netscape and Microsoft support it. Its major additions to HTML 2.0 include tables, support for applets, text flow around images, and sub- and superscripts. It does not support displaying mathematics, as HTML 3.0 did. Thus, by itself, HTML is not sufficient for technical papers.

HTML is not a programming language as programs cannot be written in it. Rather, it is a language that describes the logical structure of the document being viewed—thus the term "markup language"[9]. For instance, HTML marks paragraphs, breaks in lines, titles of documents, and types of fonts; shows links to other documents, describes the images to be shown, etc. Table 2, described later in detail, illustrates some HTML—commands are in "tags" deliminated by < and >. One surprising aspect of HTML is that it does not describe the actual "look" or appearance of the document. The same HTML on different browsers, or even on the same browser configured differently, will look differently. HTML only describes the information, and it is up to the browser to decide how to display that information[10]. Put another way, HTML describes the content and logical structure, but not the format or layout, of a document.[11]

There are many ways to create HTML; they range from the very simple to complex and sophisticated. Probably the best way to start is the simplest—writing HTML with a text processor (or any word processor that can save text files). HTML is written in a superset of ASCII known formally as ISO-8859-1, or "Latin-1"[12]. To test your HTML, use the "open file" option of your browser on your edited file (it should be saved with the suffix .htm if you are using Windows 3.1, and .html for other systems). By toggling between your browser (you may need to "reload" the image after each save with your text or word processor) and the HTML file, you can quickly try many elements of HTML. Even if you ultimately plan to use other tools, it is a good idea to try this first to get an innate feel for writing HTML.

There are many references to HTML (a count at one on-line bookstore showed almost 100 titles), so all could not be reviewed for this paper. However, Musciano and Kennedy (1996) and McFedries (1996) are useful—they give good overviews, yet are easy to read and not overly detailed. Since HTML is a creature of the Internet, there are many references available through the net; some are listed in Table 1. These are particularly useful as they keep abreast of the frequent changes to HTML, unlike books.

To learn HTML by example, many browsers allow you to view the HTML of the current page. In Netscape Navigator, the option is under "view," and is called "document source." Other browsers use similar names. This can be a useful adjunct to books and on-line sources.

There are a huge variety of HTML authoring tools. They can be broken into two categories: conversion or filtering tools those that take LaTeX, PostScript, Microsoft Word, etc. files and then convert them to HTML. The second category are HTML editors, where the author works with a special editor for HTML. In addition, some programmable editors, such as `emacs`, have extensions for HTML. Clearly, the choice depends upon the intended use and needs of the author. For instance, "Resources for Economists on the Internet" is kept in LaTeX, and `LaTeX2HTML`, a Perl program, generates the HTML. For this project, which involves a large amount of material in hundreds of subsections, `LaTeX2HTML` is a natural, and quite speedy – the last run took 6 minutes to generate 1.7 megabytes of HTML. Alternatively, an HTML editor is a good choice for a home page, which typically involves just a few pages of HTML; see Table 1 for specifics.

Given the rapid changes in the web and tools for it, it is difficult to recommend an HTML editor. Currently, for Windows platforms, "HotDog" from Sausage Software, is highly regarded. Netscape Navigator 3.0 Gold[13] edits HTML files, and it has an attractive price for educational use—zero. Finally, the future of HTML editors on Windows platforms is clouded as next versions of "productivity suites," such as Microsoft Office 97, Corel Office Professional 7, and Lotus SmartSuite 97, will include the ability to save files in HTML. In general, they will likely be out before this paper, so many users might just default to the HTML tools included with it. Table 1 also lists the sites for two HTML editors, and the two Usenet newsgroups devoted to writing HTML.

Table 2 shows an example of HTML, and Figure 1 shows the output as viewed with Netscape Navigator 3.0 running on Linux. As mentioned above, HTML commands are tags; they generally come in matched pairs. For instance, the file should start with `<html>` and end with `</html>`; matching end tags have a slash immediately after the first less than sign. HTML tags are case insensitive. The file has two parts: the head, and body, which of course are denoted by tags. This head tag only includes the title of the page, which is displayed in different ways in different browsers.[14] Note that blank lines and line returns in the HTML file have no impact on how the HTML is displayed.

This body tag uses a common extension to HTML 2.0: one for background colors. The "`ffffff`" translates as white (the default is grey); the coding is in hexidecimal from 0 to 255 for red, green, and blue, respectively. Thus, "`ffffff`" means the greatest amount of red, green, and blue, or white (in hexidecimal, ff = 255). Next, note the rather odd way that comments are written in HTML. "Headers" (which head important ideas or sections) come in different sizes; `<h1>` denotes the largest size, while the smallest is `<h6>`. To break to the next line (normally text flows uninterrupted), a `<br>` command is used. As described in the example, `<p>` indicates paragraphs, and a common extension to HTML 2.0 lets one center a paragraph. Another common tag is `<hr>` for a horizontal ruler, and two ways of the many ways of marking text itself as described in the example.

One key feature of HTML is linking to other sites on the Internet. In this case, a link is made to a useful site for summary macro data. Note the beginning and ending "anchor" tags. The href attribute denotes the URL[15] of the site, and the text between the URL attribute and the closing anchor tag is shown on the page as a clickable link—a mouse click on this entry will bring in this page and shift the user to this site.

The next set of HTML shows an unnumbered list; a `<li>` denotes each list item (besides `<p>`, this is one of the few tags that needs no ending tag). The next bit of HTML shows how

Table 1: Internet HTML Tools and References

Short References and Guides

NCSA Beginner's Guide to HTML A short introduction to essential HTML.
`http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html`

Bare Bones Guide to HTML Lists all HTML 3.2 and Netscape tags.
`http://werbach.com/barebones/`

General References and Guides

The Web Developer's Virtual Library Many links and extensive reference information.
`http://www.stars.com/`

Web Mastery Resource Lists Many links and extensive reference information.
`http://union.ncsa.uiuc.edu/HyperNews/get/www/html/guides.html`

HyperText Markup Language (HTML) General info from the World Wide Web Consortium.
`http://www.w3.org/pub/WWW/MarkUp/`

Yahoo Section on HTML Links to many HTML sites.
`http://www.yahoo.com/Computers_and_Internet/Software/Data_Formats/HTML/`

Guides to HTML Editors and Filters

World Wide Web and HTML Tools Info on tools from the World Wide Web Consortium.
`http://www.w3.org/pub/WWW/Tools/`

Producing HTML Documents Portion of the "World Wide Web FAQ" by Thomas Boutell.
`http://www.boutell.com/faq/htprod.htm`

The Consummate Winsock Apps List – HTML Editors Part of a leading guide to Windows applications.
`http://www.icorp.net/stroud/html.html`

Specific Editors

Sausage Software Award-winning HTML editor for Windows.
`http://www.sausage.com/`

Netscape Communications Leading browser company; also has HTML editors.
`http://home.netscape.com/`

Usenet Newsgroups

`alt.html` Original HTML newsgroup; relatively low volume.

`comp.infosystems.www.authoring.html` "Official" HTML newsgroup on writing HTML; high volume.

Figure 1: Displayed HTML

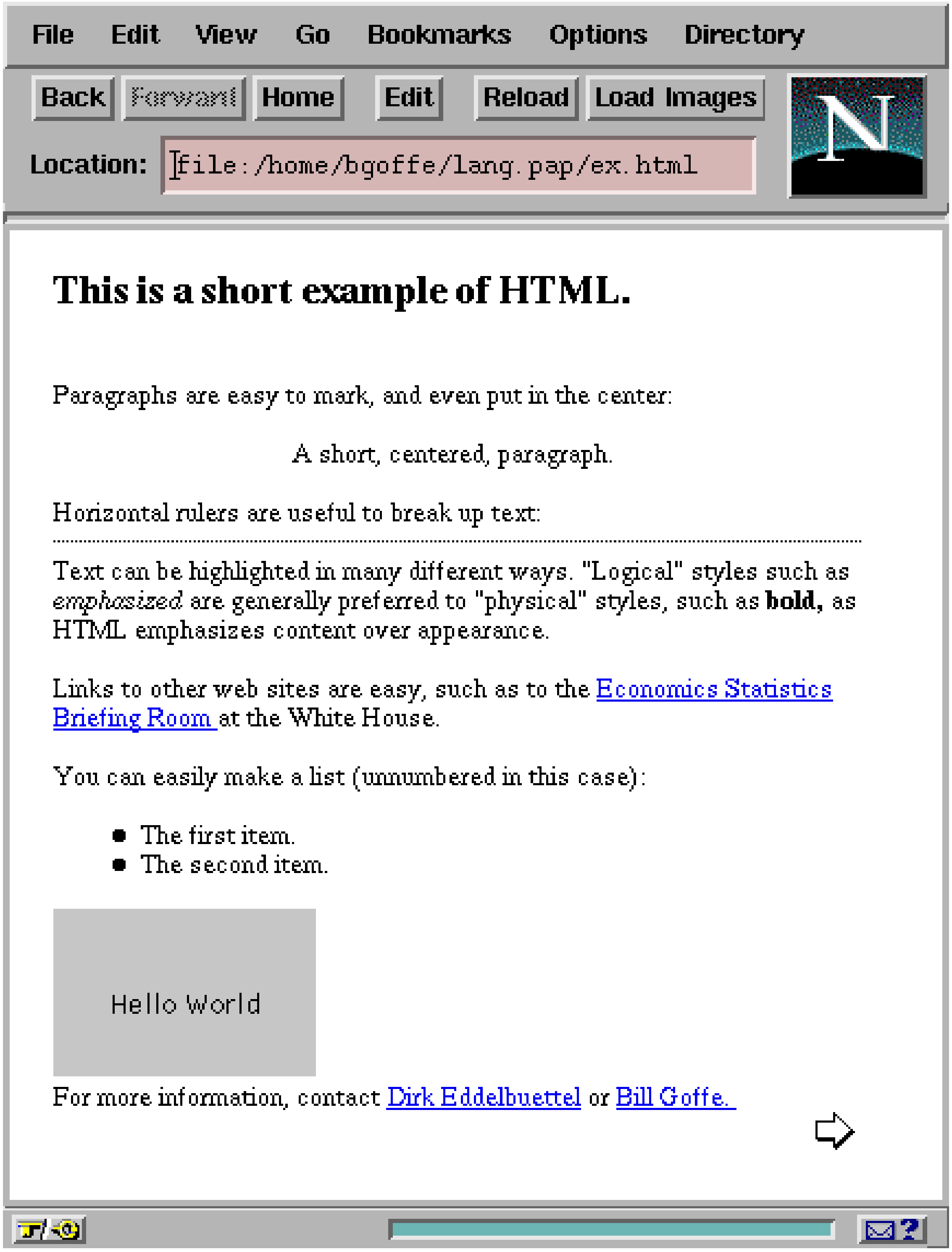

Table 2: HTML Example

```html
<html>
<head>
<title>HTML Example</title>
</head>

<body bgcolor="ffffff">
<!-- Comments in HTML have rather odd delimiters. -->

<h1>This is a short example of HTML.</h1><br>
Paragraphs are easy to mark, and even put in the center:
<p align=center>
A short, centered, paragraph.

<p>
Horizontal rulers are useful to break up text:
<hr>
Text can be highlighted in many different ways. "Logical" styles such as
<em>emphasized</em> are generally preferred to "physical" styles, such as
<b>bold,</b> as HTML emphasizes content over appearance.

<p>
Links to other web sites are easy, such as to the
<a href="http://www.whitehouse.gov/fsbr/esbr.html">
Economics Statistics Briefing Room </a> at the White House.

<p>
You can easily make a list (unnumbered in this case):
<ul>
<li> The first item.
<li> The second item.
</ul>

<applet code="HelloWorldApplet.class" width=120 height=80> </applet>
<br>
For more information, contact <a href=mailto:edd@ehess.cnrs-mrs.fr>
Dirk Eddelbuettel</a> or <a href=mailto:Bill.Goffe@usm.edu>Bill Goffe</a>.

<a href="next.page.html">
<img border=0 align=right src=arrow.gif></a>

</body>
</html>
```

a Java applet can be imbedded in HTML; details on Java are in the final section of the paper. Note the `<br>` tag to put the next entries on their own line. When clicked, the "mailto" anchor activates the user's browser's e-mail program with mail addressed to the addressee given by the href attribute. Next, an image, from the file "arrow.gif" serves as a navigation icon—clicking on this icon brings in another HTML file: "next.page.html." There are two HTML 2.0 attribute extensions here. The "border=0" attribute, means that no unsightly border is drawn around this GIF. The "align=right" attribute places the GIF on the right side of the page[16]. The file ends with two closing tags for the body and the HTML.

To make information written in HTML available to others over the Internet, the HTML file or files must be placed on a web server (technically known as an http server). Describing how to run a web server is beyond the scope of this paper; we merely note that there are many web servers on the Internet—one count in early November 1996 found some 460,000. Many universities run web servers, so there should be easy access for academics, and local internet service providers also make servers available.

3 PDF

The software firm Adobe, well known for its desktop publishing and layout products, including PostScript, Illustrator, and Photoshop, invented the PDF (portable document format) file format[17]. As its name suggests, this file format is designed for displaying documents on different platforms. The philosophy of PDF is very different from HTML; in fact, it is almost the reverse. HTML describes the content and logical structure of the document (the material itself, and tags for sections, paragraphs, titles, etc.), but not the layout or format of information. PDF files describe the document's layout, but not its logical structure. Thus, with PDF, the document will look the same to all readers on all platforms, but titles, headings, sections, images, etc are not denoted[18]. One well known user of PDF is the U.S. Internal Revenue Service (IRS). At their web site, they offer their innumerable forms to the public in PDF format—it is a natural for documents that must appear and then print identically for all users; this is currently impossible with HTML. For academics, PDF is ideal for working papers—it will faithfully reproduce all fonts, equations, and tables. PDF files can be thought of as "digital paper"—they show information exactly, but provide no special information to the user or program about what it is displaying.

PDF files are an outgrowth of PostScript, the dominant high-end "page description language." These languages describe the content of a page: where text and graphics are located, what sort and size fonts are used, the actual graphic, etc. PostScript is very flexible, as it is actually a programming language[19]. It is also "platform and device independent," so PostScript files can be used on different platforms. For instance, a PostScript file might be produced on an Apple Macintosh, and printed on a local printer attached to an IBM PC. It could then be e-mailed (PostScript is plain text) to a commercial printer and printed with commercial typesetting equipment with much higher resolution. These features have made PostScript the de facto standard for advanced publishing[20].

Unfortunately, PostScript suffers from one key limitation: it assumes that each user has the author's fonts. In addition, PostScript files often are rather large. PDF solves these limitations in several ways. First, PDF files include the fonts (or a close facsimile) in the file. Second, PDF uses a subset of PostScript that reduces file size. Finally, there are a variety of compression options available. With these features, Adobe's track record in desktop publishing, their marketing (including making PDF readers freely available and easily integrated with browsers), PDF has become the de facto standard for displaying information through the Internet that requires precise layout.

There are several ways to read PDF files. The most common is through Adobe's freely available PDF Reader software, which is offered for many platforms (Adobe generates revenue from their PDF family by selling the products that generate PDF files). Readers can be obtained free of charge from the Adobe web site. In addition, there are two freely available non-Adobe solutions: Ghostscript and xpdf. Ghostscript reads and prints PostScript and PDF files, and with Ghostview, GSview, or GV displays them on a monitor. Versions are available for many platforms. The xpdf program runs on X Window systems; it displays PDF files or converts them to PostScript (typically for printing). When a PDF file is selected with the appropriately configured web browser, these programs are automatically started to view the PDF file. As mentioned before, this is only automatic if the browser is set up correctly. URLs are listed in Table 3.

There are several methods of generating or creating PDF files. As one would expect, Adobe leads the way, but there are alternatives as the PDF format is publicly available. As PDF's popularity increases, more non-Adobe products are likely to arise[21]. Acrobat 3.0, scheduled to ship in mid-November 1996, includes Reader, Exchange, Capture, Catalog, Search, PDFWriter, and Distiller. This package, available for many platforms, will interest most users interested in creating PDF files. Reader reads PDF files (it is also freely available through the Internet at the Adobe web site), while Exchange does that and more: you can also add notes to files, save files (unlike Reader), and add hyperlinks. Capture lets one create PDF files from scanned documents (thus, one can deal with an "installed base" of paper documents), while Catalog and Search create and read indices of PDF files. Finally, both PDFWriter and Distiller create PDF files. PDFWriter is a bit easier to learn and use, but is more limited. It operates as a printer driver for popular programs: to use it, you print the file to disk through this program, and the resulting file is in PDF format. Distiller is a more flexible package. It operates as a translator—it takes PostScript files as its input, and outputs PDF files. As almost any program can generate PostScript, this alone makes it more flexible than PDFWriter, which is available for only a few programs. While it takes a bit more effort to learn, Distiller offers many useful options, including numerous ways of compressing the file and its components, and inclusion of encapsulated PostScript files. In short, Distiller is more robust and feature laden than PDFWriter. Information on the Acrobat family can be found at the Adobe web site, whose URL is listed in Table 3.

There are non-Acrobat methods of generating PDF files. Some desktop publishing systems and graphics programs, including Adobe's PageMaker, Illustrator, and FrameMaker can save files in PDF format. The Taiwanese software firm Zeon also offers a family of PDF products; they offer support for Chinese language PDF files. Apparently the only freely available method of generating PDF files are the latest versions of Ghostscript, which includes ps2pdf. While its current capability is somewhat limited (it doesn't compress the files, and for best results you must use PostScript fonts), one suspects that with time, it will gain features. The URLs for these products are in Table 3. Finally, as PDF gains popularity, and with the PDF file format being publically available, one should expect more programs that generate PDF files.

The first PDF file format and readers were not optimized for use on the Internet; when PDF was developed, the web was in its infancy. However, the most recent reader, 3.0 (earlier named "Amber"), tightly integrates with browsers. Older PDF readers are categorized as "helper applications:" when properly configured, they open up a new window to read the selected PDF file. The 3.0 Reader is a "plug-in:" the reader program opens in the browser itself, and not in a separate window. This is obviously less distracting to the user, and there is no difference between reading a page written in HTML or PDF. In addition, newer PDF creators can produce PDF files that can be viewed one page at a time—older PDF files required that the entire file be on the user's computer before any of the file could be viewed. With these advances, and since PDF files can include hyperlinks, PDF can be viewed as a replacement to HTML. Practically, this is not likely as PDF readers and generators are not widely deployed, but these additions to the PDF family offers all the major functionality of HTML: color graphics, text, and hyperlinks tightly integrated with a browser.

Table 3 lists a number of resources for PDF. Note that some are corporations or others that offer PDF software, while others are guides or references (including mailing lists and a Usenet newsgroup). While there are at least six books devoted to PDF, Page and Holm (1996) is quite useful; among other things, it offers many suggestions for those planning sites with PDF material.

4 Java

For anyone with at least a minimal interest in computers, it was close to impossible *not* to have come across a publication which featured Java in its title during most of 1996. Numerous magazine cover stories were devoted to this new computer language, and an impressive number of books, including many in newly created series on Java, have recently been published[22]. In this section, we will highlight some important aspects of Java. Unfortunately, available space limits the scope of this survey; however, pointers to more detailed references are provided at the end.

Much of the interest in Java stems from its uses in network computing and, in particular, on the World Wide Web part of the Internet. Java can be used to make the web more dynamic, accessible, user-friendly, and interactive. However, as detailed below, Java is a general-purpose programming language that can be used for far more than the early applications of spinning graphics on web pages. For example, according to a (at the time of this writing, recent) article in *The Economist*, Corel is close to launching an application "suite" of word processor, spreadsheet and graphics written entirely in Java, and can therefore be used on all computer systems with a Java Run-Time System (see below) as for example the ones that are built into Web browsers such as Netscape Navigator or Microsoft Internet Explorer. Moreover, a new class of computers will be soon be available in stores: the so-

Table 3: Internet PDF Sites and Resources

Readers and Generators

Adobe Corporation This site offers their free Reader and material on other PDF products.
`http://www.adobe.com/acrobat`

Ghostscript, Ghostview and GSview Freely available software for PostScript and PDF viewing and printing.
`http://www.cs.wisc.edu/~ghost/`

Zeon Corporation PDF software with extensions for Chinese.
`http://www.zeon.com.tw/`

References, Aids, and Guides

comp.text.pdf FAQ The "Frequently Asked Questions (and Answers)" for the PDF newsgroup.
`http://www.vertec.com/info/info.htm`

Emerge Corporation Offers add-on products for PDF, and also a large amount of information on PDF.
`http://www.emrg.com/`

Acropolis An on-line magazine devoted to PDF.
`http://plaza.interport.net/acropolis/`

comp.text.pdf Usenet newsgroup devoted to PDF.
`http://plaza.interport.net/acropolis/`

PDF-L and CAPTURE-L Two mailing lists devoted to PDF and Acrobat Capture.
`http://www.emrg.com/ed_lists.html`

The Acrobat Talk Mailing List Another mailing list devoted to PDF.
`http://www.blueworld.com/lists/acrobat/`

called 'Network Computers' (NC) that will use a Java operating system. These machines are expected to be relatively simple with just enough built-in capabilities to boot. All software will be downloaded on demand from the network. This makes the NC attractive for both the corporate (where there are typically a large number of networked personal computers) and the domestic markets, and forms a threat to Microsoft's dominating position in both these markets.

Java captures considerable attention because it allows for compact, expressive, object-oriented code, just as C++. Moreover, it provides a mechanism for flexible, secure and transparent applications on the World Wide Web. Finally, Java is also a convenient way of writing large applications across heterogeneous platforms because of its seperation between the "virtual" Java Machine and the actual Run-Time System as we discuss below.

For economists, Java is very a promising tool for both teaching and research. An excellent example of the use of Java for an interactive microeconomics 'textbook on the Web' is oo_Micro (which can be visited at the URL `http://medusa.be.udel.edu/WWW_Sites/oo_Micro.html`). It uses Java to enable interactive modifaction of constraints, endowments and preferences etc., and it updates the corresponding graphs immediately. Another example is the statistics department at UCLA (`http://www.stat.ucla.edu`) where Java is used for interactive statistical calculators and the display of different probability functions.

Java is the successor to the Oak language that was developed at Sun Microsystems by a team led by James Gosling. Oak was initially targetted as a computer language to program consumer electronics devices such as toasters, microwave ovens, and set-top boxes for interactive television. This implied requirements such as a clear seperation from hardware, as the programs have to operate on different and changing computer chips. It also required that the language be small and reliable. With the advent of the World Wide Web, Sun Microsystems realised how this language concept fits the distributed and heterogeneous nature of the World Wide Web. The architecture-neutral Java language, through all its design goals, suits the Internet extremely well: different computers can download the same Java programs and execute them directly (as we will disuss in more detail below).

Sun Microsystem reacted promptly to this opportunity and developed 'HotJava', the first Web browser which could download and execute Java 'applets'. These are Java programs that can be embedded into another context, for example a Web page. Through applets, the Web becomes interactive and dynamic as program execution is provided directly[23]. Following the success of the HotJava browser and the first releases of the Java Development Kit (JDK) by Sun, programmers started to experiment with Java and to add Java content to the Web; pointers to some applet collections are provided below.

Sun Microsystems created the Java language and controls its development. Licences (which have to be acquired for commercial development of Java applications) have been given to many firms in the computer industry, including Microsoft and IBM. However, no license is required to write and distribute most Java applications and applets[24].

So what is the Java language like? To anyone with experience programming in either C or C++, Java looks very familiar. From C, Java inherits a compact, expressive style. But unlike C++, Java does not try to be backward compatible with C which renders C++

(in the eyes of some Java advocats) unnecessarily complex. Java is both procedural as standard languages as C, Pascal and Fortran, and object-oriented as C++. If we think about programming in terms of languages, traditional (procedural) approaches specify what the computer should do through sequences of intructions; computer programs are then seen as *verbs*. On the other hand, the object-oriented approach of software development focusses on *nouns*: the programmer decides on the types of objects a program should comprise, and then designs actions for these objects. Object-oriented programming uses entities, the objects, that have both a state (as they contain data) and behaviour (as they contain so-called 'member' functions to alter their data and to interact with other objects). As other object-oriented languages, Java supports classes (which are the means to define objects) and single class inheritance (to pass characteristics from one abstract object to the next). Unlike C++, Java does not support operator overloading (a means to redefine standard operators such as '+' for operations between objects).

Java is completely class-oriented which means that *all* data have to be encapsulated; and that in turn means no global variables are permitted. Leightweight processes, also known as threads, are supported, which enables programs to have different parallel subprocesses in a synchronised manner. Unlike C, Java does not have header files and does not need a pre-processor, something C++ maintained for compatibility with C. In Java, modules are directly imported, and code and definition are in one file. For memory management, Java follows other object-oriented languages (with the notable exception of C++) and provides 'garbage collection'. This means that the system—and not the programmer—is responsible for of allocating, regaining, and freeing memory for dynamic objects.

Consequently, Java does not have pointers. The C and language needs them for a cumbersome call-by-reference mechanism (which means that variables passed to a subroutine can be changed persistently by the subroutine) and for dynamic arrays. Both call-by-reference and dynamic arrays are available directly in Java. Pointers therefore are not needed; this eliminates the single most common cause of programming mistakes. Java objects use reference semantics: this provides for the efficiency of pointer use with the safety of call-by-value. Base types as integer or character types use value semantics.

Java also provides run-time checking of code: it is impossible to derefence a pointer outside a segment, use de-allocated memory, or objects of the wrong type because of the Bytecode verifier (described below). Finally, Java provides a standard application-programming interface (API) for basic file input and output, networking (including HTTP), dynamic base contructor classes, graphics, and graphical user interfaces.

Java as a language is tied to the Java Virtual Machine (JVM), which as the name implies is an abstract concept[25]. For traditional programming languages, a compiler usually turns source code (written in a compiled language as C, C++, Pascal or Fortran) into assembly code, which the operating system can execute directly. The generated code therefore depends on both the hardware and the operating system: programs that are created from C++ source code on, say, a Pentium-based computer running Linux will not run on either a Pentium running OS/2, or a DEC Alpha running Linux. This makes programming for different hardware platforms, and operating systems, cumbersome; C programs intended for use on different computers often have large blocks of mutually exclusive code segments that are activated during conditional compilation by directives given to the pre-processor. Java does much better by providing two different layers. The abstract description of the Java Virtual Machine is completely seperated from the implementation hidden in the Java Run-Time System. Only the Run-Time System, which actually executes Java code, knows about the underlying hardware, and handles machine-specific issues. Therefore, only the Run-Time System is needed to execute Java programs on a given hardware platform.

The Java Run-Time System (JRTS) executes the Java Bytecode, usually after interpretation. It is the only component of a Java implementation which knows about the particular system on which it is running. The Java programs themselves are seperated completely from hardware—which in turn makes the program independent of these hardware aspects. The JRTS also loads compiled classes and executes security checks. The easiest way of obtaining a Java Run-Time System is to simply install a recent version of a popular Web browser as Netscape Navigator or Microsoft Internet Explorer which each have a JRTS built-in. The combination of the Java Virtual Machine and the Java Run-Time System makes it possible to "write once, run anywhere". This key feature of Java promises to considerably reduce development and maintenance efforts for software designed for multiple platforms.

As Bytecode is (typically) interpreted, the execution speed is much slower than for compiled C code. However, more and more Java run-time systems come with a Just-In-Time (JIT) compiler which transforms Bytecode into native assembly code. In late 1996, speed increases of a factor of up to twenty have already been reported. Wider availability of Just-In-Time compilers for Java will give programmers the speed of C. This performance comes without any portability sacrifices as the Just-In-Time compiler is part of the Java Run-Time-System and not of the Java Developement Kit (or another compiler as Guave).

Security concerns are addressed at different levels. The Java Virtual Machine itself provides passive security. Consider a (traditional) compiled program. As its code is native to the hardware, it can easily exploit known weaknesses on this hardware platform. Bytecode, on the other hand, is architecture neutral and ignorant of the hardware it is running on so that it cannot exploit this hardware. The Class Loader in the Run-Time System further enhances security by isolating classes: those built into the RTS from those that are local to the account; both of these from those of other accounts on the same machine, and other machines. Foreign classes cannot 'disguise' and thereby shadow built-in classes. The Java Security manager, a standard class provided in the API, implements the security precautions for all untrusted programs, which are generally those downloaded over the Internet. No read or write access to files is given to such a Java program and no network connections can be established to hosts other than the one from which the applet itself was downloaded. Finally, the browser often allows users to chose among different security levels to limit (or deny) access for untrusted applets.

To write actual Java programs and applets, Sun offers the Java Development Kit (JDK) for Sparc and Solaris workstations, Windows 95 and NT, MacOS 7.5 on PowerMacs and 68020, 68030 and 68040 Macs. Ports have been made by others to Silicon Graphics IRIX system and Linux. IBM ported the JDK to OS/2 and AIX. The Open Software Foundation

(OSF) has ports to Unixware, HP/UX and Digital Unix.

As an alternative to the JDK, programmers can use the ‘Guave’ compiler for Java, which has been released under the GNU General Public Licence (GPL); see `http://http.cs.berkeley.edu/~engberg/guavac` for more information. On a Linux system, the Guave compiler `guavac` runs much faster than the JDK. Moreover, a Java compiler called ‘EspressoGrinder’ has been written in Java itself. This Java compiler is therefore available on any system with a Java Run-Time System; see `http://wwwipd.ira.uka.de/~espresso/`.

Java programs are created by using a compiler (for example from the JDK, or guavac) to compile a Java source file (typically with extension `.java`) into a file with extension `.class`. The latter file contains the platform independent pseudo-assembly *‘Bytecode’*[26]. A Java Run-Time System (RTS) then interprets and executes the file. The (concrete) Run-Time System, which emulates the (abstract) Java Virtual Machine, is the only program that needs to be ported onto a new computer platform. Once a Run-Time Systen is available on a given platform, *all* Java programs can run on that hardware platform. While applets are currently the pre-dominant form of Java programs, stand-alone programs also can be created. As an example, consider the one shown in Table 4 which honours a longstanding tradition and prints out the string ‘Hello World!’.

Table 4: Java Example

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

The example contains a declaration of one class which has the same name as the file it is in: HelloWorld. The class contains one member function: the main() function, which uses the println() member function from the System.out class to display the text string. Java uses the convention of choosing the name of a class for the file that the class is described in; in this case ‘HelloWorld.java’. To compile this file into Bytecode, we simply call a Java compiler as in `guavac HelloWorld.java` which then creates a file HelloWorld.class which can be executed.

In the second Java example, taken from Flanagan (1996, p. 86), displaying ‘Hello World’ is now implemented as an applet. Two import statements make the predefined applet and windowing classes of the Java API available to this class. The text is displayed through the Graphics object using the drawString member function. The HTML example above already showed how to integrate this applet into a Web page via the `<applet>` tag.

As Java is itself so closely linked to the advent of the World Wide Web, it is not surprising that extensive documentation on Java can be found on the Internet. Table 6 lists several of

Table 5: Java Applet Example

```
import java.applet.*;
import java.awt.*;

public class HelloWorldApplet extends Applet {
    public void paint(Graphics g) {
        g.drawString("Hello World", 25, 50);
    }
}
```

these sites. Among the many books on Java, Flanagan (1996) stands out as it serves at the same time as a tutorial on Java, and as a reference guide to all Java classes, variables, and methods; however, it might not be the easiest book for beginners.

Table 6: Internet Sites and Resources for Java

Sun Microsystems This site contains the latest published versions of official Java information and software as the JDK.
`http://www.javasoft.com`

Java Mailing Lists Information about several Java related mailing lists.
`http://www.javasoft.com/mail.html`

Gamelan The largest collection of Java applets.
`http://www.gamelan.com`

Yahoo Section on Java Links to many WWW sites with Java information and content.
`http://www.yahoo.com/Computers/Languages/Java/`

The Java Online Bibliography General articles about Java and Java applications.
`http://www.rpi.edu/~decemj/works/java/bib.html`

Cup O'Joe Java Shop Another useful site with links to more Java information.
`http://www.cupojoe.com`

Charles Prineas Java Links Yet another useful site with links to more Java information.
`http://www.prineas.com/Links/Java/index.html`

Java Newsgroups Several Usenet discussion groups devoted to Java.
`comp.lang.java.*`

5 Conclusion

This paper examined three different technologies for displaying information on the Internet, and interacting through it. Simple HTML is quite easy to generate, and its use on the

Internet is ubiquitous as it is integral to the World Wide Web. However, it suffers from a key drawback in the eyes of many: it does not allow precise control of format or layout, and by itself, cannot display mathematical expressions. Thus, while HTML is ideal for "home pages" and other material that do not require precise formatting, it is not suitable for material such as working papers or publications. Fortunately, there is a solution: PDF. It was created to maintain the precise layout the document's author had in mind. It is easy to install and use a PDF reader, and only slightly more work to generate a PDF file. Thus, it is ideal for working papers or an on-line journal. Finally, Java offers many possibilities. It is a general-purpose programming language with a strong influence of C and C++. As Java programs are compiled into architecture-neutral Bytecode, they can be executed on any computer with a Java Run-Time System. Java therefore enables programmers to provide interactive applications in a flexible and convenient manner through the World Wide Web.

Notes

[1]The transfer of information uses HTTP (hypertext transfer protocol)

[2]The file name often ends with .html; or .htm on servers which cannot support extensions of four characters.

[3]These programs are typically classified as applets, plug-ins, or helper applications. The browser must be correctly configured for them to be called; this configuration is not (yet) automatic.

[4]But, to understand what these programs do, it is quite useful to know a bit of HTML and its philosophy.

[5]As described earlier, HTTP communicates between browsers and web servers and transfers files of HTML and other information.

[6]Their website at `http://www.w3.org` provides a wealth of information on the consortium.

[7]Like all formal Internet standards, the Internet Engineering Task Force (IETF) approved this standard; this one is referred to as RFC 1866.

[8]But not including "frames," which sets up a web page as different documents.

[9]HTML 2.0 and the proposed 3.2 are defined as a type of a more powerful, but much less commonly used markup language, SGML (standard generalized markup language).

[10]Economists might have come across this distinction between logical structure and the visual appearance of a document with LaTeX, a commonly used computer typesetting system. Like HTML, LaTeX describes the document's logical structure, and while users can change the appearance of the document, LaTeX does produce a default style of output.

[11]This will change with "style sheets," which describe the format of an HTML page. As of late 1996, style sheets are still being standardized. W3C has a standard for style sheets, called "Cascading Style Sheets Level 1" (CSS1), but it is not clear if Netscape will support this proposed standard.

[12]If your system only supports ASCII, your only limitation is that you will not be able to use some non-English characters.

[13]While this is a special version of Navigator 3.0, future versions of Navigator will apparently include an HTML editor.

[14]Here, it is in the border of the window, which is not pictured.

[15]A URL is the Uniform Resource Locator, which describes three things: the type of protocol, (HTTP, FTP, etc.), the host computer with the information, and the location of the file on that computer. In this

case, the protocol is http, (note the :// deliminator), the host computer is www.whitehouse.gov, and the location of the HTML file on that host is fsbr/esbr.html.

[16]One commonly seen modification with GIFs are animated ones, or a sequence of gifs that simulate motion.

[17]The term Acrobat is often used, but strictly speaking Acrobat is a family of Adobe products for creating and viewing PDF files.

[18]However, Adobe is working on adding this information to PDF files. Since layout facilities are being added to HTML with style sheets, it appears that someday that PDF and HTML will reach roughly the same functionality. But, that day is still some ways in the future.

[19] The PostScript langueage can even be used to invert matrices.

[20]While we haven't checked, it is likely that many journals use PostScript in their production process.

[21]In much the same way, the PostScript specification is publically available and alternatives to Adobe are on the market. However, as they do with PostScript, Adobe intends to stay ahead of other vendors with PDF products.

[22]Java is jokingly referred to as 'fully buzzword compliant'; Sun Microsystems earned this title by introducing the language in a White Paper as *'Java: A simple, object-oriented, distributed, interpreted, robust, secure, architecture neutral, portable, high-performance, multithreaded, and dynamic language'.*

[23]This offers much more flexibilty than the current method of interaction over the web: the user selects items from a "form" written in HTML, and the remote server responds using a system known as "Common Gateway Interface" (CGI) which allows execution of programs which communicate with the user through the CGI interface. As the CGI specification is language independent, C or Fortran programs can be executed; however Larry Wall's Perl language is most often used. We are not covering CGI in more detail as this paper focusses on languages, rather than interfaces.

[24]Legal details are provided at `http://www.javasoft.com/java.sun.com/source.html`.

[25]It is worth recalling that Wirth's Pascal compilers of the late 60s already compiled into a "pseudo code" that is conceptually similar to Java's Bytecode. The compiler output, the so-called p-code, was then executed by a "Pascal virtual machine", which was named the "p-code interpreter". UCSD Pascal and its derivate Apple Pascal of the mid-80s are implementations of this concept.

[26]Bytecode is described in the specification of the Java Virtual Machine (JVM) which is itself described at `http://java.sun.com/doc/vmspec/html/vmspec-1.html`.

References

Flanagan, David (1996). *Java in a Nutshell.* Cambridge, MA: O'Reilly and Associates.

McFedries, Paul (1996). *The Complete Idiot's Guide to Creating an HTML Web Page.* Indianapolis: Que Publishing.

Musciano, Chuck and Bill Kennedy (1996). *HTML: The Definitive Guide.* Cambridge, MA: O'Reilly and Associates.

Page, Bruce and Diana Holm (1996). *Web Publishing with Adobe Acrobat and PDF.* New York: Wiley Computer Publishing.