

Technology Briefing report on Internet Technologies

By Jérôme Guionnet, Cap Gemini Innovation

1. OVERVIEW	1
1.1. Introduction	2
1.2. An Overview of the Internet	2
1.2.1. Client Browser Software	2
1.2.2. Internet Search Engines	2
1.2.3. Server Software	2
1.2.4. Web Sites Utilities	3
1.2.5. Web Site Development Software	3
1.2.6. Linking Company Databases	3
1.2.7. Linking Legacy Applications	5
2. ROLE IN EVOLUTION PROCESS	6
3. RELEVANCE TO RENAISSANCE	7
3.1. Web vs. Client/Server: Rapid Application Deployment	7
4. AVAILABLE SUPPORT	7
5. MATURITY ASSESSMENT	8
6. INAPPLICABILITY	8
7. FUTURE DEVELOPMENT	9
7.1. JDBC Future Plans	9
7.2. ActiveX	9
8. OTHER COMMENTS	9
8.1. Comparing Netscape One and Active X	9
8.1.1. Plateform- dependent	10
8.1.2. Windows-centric	10
8.1.3. Insecure	10
8.1.4. Proprietary	10
9. REFERENCES	10

1. Overview

(Description of the technology in general terms.)

1.1. Introduction

The Web stirs the imagination. 30 million users world-wide can access any page of information from anywhere in the world at the click of a button. Global information access is suddenly upon us just five years after the Web was an idea in a scientist's mind. The publicity is unprecedented and the growth phenomenal. 100 new web sites per day are joining the Web. It is estimated there will be 200 million users by the year 2000. A single world-wide market to sell and support goods, services and information available to anyone with a computer.

1.2. An Overview of the Internet

To define the Internet in more concrete terms, we will now consider all the software which are used within a simple Internet/Web system, we will describe them very quickly and will consider which are relevant for Renaissance.

1.2.1. Client Browser Software

An internet browser sits on a client machine and gives its user access to the Web. A browser can either be an independent software package that can sit on top of various operating systems, or it can be embedded into an operating system. Similarly, a browser can either be an all-purpose browser or it can be specialised in one internet protocol. Thus, until recently, Netscape's Navigator was usually considered a Web (HTML) browser while Eudora was generally considered an e-mail (SMTP) browser. From release 2.0 Netscape's Navigator provides support for all the popular Internet protocol (Web, e-mail, news groups) and we can expect that the various specialised browsers will soon disappear.

The most popular browser is Netscape's Navigator product, which supposedly has over 85% of the market. Microsoft is working very hard to establish its own browser, Internet Explorer.

Microsoft's browser is currently independent of MS Windows, but will be built into OS in future releases. There are lots of other small companies offering browsers, but they are likely to disappear in the next 2-3 years and the battle will probably be between Netscape and Microsoft.

This part does not seem to be relevant for Renaissance, but we must follow which strategy the 2 big will adopt in the future. The fight for the control of the internet/intranet is beginning and what we can predict is that they will include technologies inside their browser in order to win the war. Netscape and Microsoft both incorporate Java and Javascript technology into their Web browsers. However, this is not the only solution and they planned to incorporate different solutions to be linking to other applications.

Microsoft has already announced that it will include a VBScript engine in its web browser (available nowadays) and that the VBScript engine will support activeX components. In effect, Microsoft will support its proprietary OLE/COM infrastructure within its Web browsers and servers.

At the beginning of the year 1996, the OMG emitted the desire Netscape would provide stubs for CORBA and support for OpenDoc in its Web browsers and servers, and we could suspect Netscape will. Sun and Netscape are close after all and Sun has already linked Java with CORBA (JOE) and will necessarily link it to OpenDoc components now that the OMG has adopted OpenDoc as a part of the overall CORBA standard. Moreover, Netscape knows that it is in a fight with Microsoft, and undoubtedly assumes that the enemy of its enemy is its friend. During the summer, Netscape of course had adopted this strategy in the product named Netscape One.

We must follow the war for Renaissance and certainly in a first time examine both strategies.

1.2.2. Internet Search Engines

The Internet offers so much that an individual can quickly become overwhelmed. A number of vendors have developed indexes or search engines that make easy to find things on the internet or in an intranet. Some are incorporated in the browsers. This will be an interesting area and companies who provided search engines in the past have usually built a bridge with the internet and except if one is integrated in Renaissance's application, it is not relevant for us.

1.2.3. Server Software

A corporate Internet/Intranet site needs software to manage the flow of messages from users or employees to this site. This is the role of the server software. Lots of different vendors offer server software and both Netscape and

Microsoft are rushing into this arena. Microsoft will clearly build server software into the NT operating system within a year or two. There will be a lots of competition in this niche and predictably, within a few years, only a few large companies will be left. Renaissance project should avoid using server software which propose feature not normalised or supported only by a few server software, but the biggest.

1.2.4. Web Sites Utilities

A number of companies are offering specialised utilities to facilitate internet site management and communications as for example: companies focused on facilitating credit card or money transfers via the internet, companies offering encryption services and tools that transmit voice over the internet, others are offering automatic translation services.

There are a lots of companies involved in lots of different niches. If one of these softwares is included within a renaissance's application, we should examine it, else there are of no interest for the renaissance's project.

1.2.5. Web Site Development Software

This includes software tools for developing web sites and for managing the sites once they are up and running. It includes simple tools for developing a few static Web pages, tools linking Web pages to new databases in order to make them more dynamic, languages for embedding small applications (applets) into web sites.

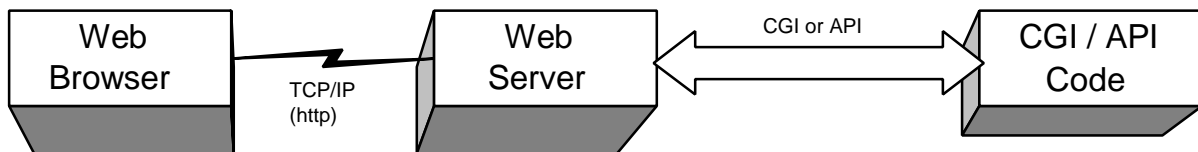
These software should be used for new renaissance's application and we will have to test and use some of them.

1.2.6. Linking Company Databases

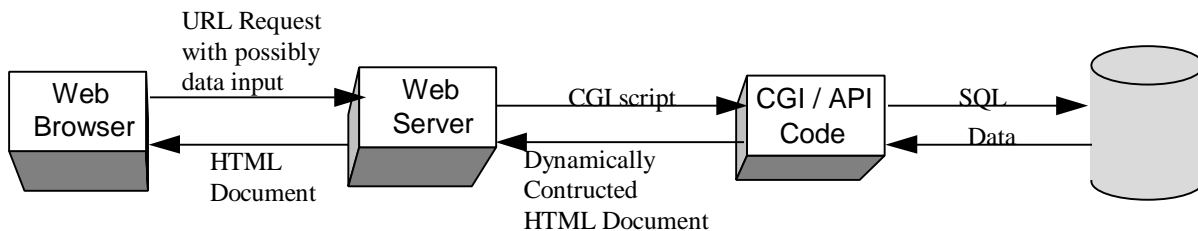
If you link items on a Web page to a database, you can use the database to update the web page (this what is done within the renaissance web server) . There are three main means to do that and so we will detail each now.

1.2.6.1.CGI, Server API

The HTTP server provides a method for running external programs from within an HTML document. This method is called the Common Gateway Interface (CGI). The server can pass information to the CGI program (if needed) and get back any results returned by the program. This method allows a user to enter information needed by the CGI program, then simply push a button to activate the program. The program processes the information, performs any other necessary functions, and returns results either in HTML format or in straight text. The HTTP server accepts the results and processes them as needed.



The CGI method of program interaction lends itself particularly well to the task of getting information from a database. Database vendors provide many different methods of accessing their products. Such methods include a programming API, a command-line interface, or a menu-driven interface. CGI's flexibility allows the use of any access method, and even the use of multiple databases on multiple servers.



CGI programs can be of practically any type, including sed, awk, perl, shell scripts, or executable programs. CGI programs just need to be able to either read data from input, or access environment variables for the data. Once

the data is read, the program executes, and then generates results either enclosed in HTML "tags" (formatting statements), or as ASCII text. The http server interprets the results and displays them.

While CGI programs can be of practically any type, most programmers seem to be migrating to the Perl programming language. Perl's text processing abilities and versatility as well as its cross-platform support make it a natural choice for CGI scripts. Moreover, it is free and interpreted so very easy to use but for rather big scripts provides poor performance.

An SQL select statement (insert, update, or delete) can be passed in pieces and the CGI program can assemble the statement, submit it to the database, retrieve and format the results, and pass the results back to the HTTP server. Using HTML tags, the data can be displayed in whatever format is needed.

Server's API offer the same kind of facilities, it allows you to add kind of pre and post processor when the server treats a demand. These kinds of pre and post processor are usually written in C language and are dynamically linked to the server. They provide faster access.

Tools and libraries are available on the market to help integration of legacy database using this technology, they will be discussed in next chapters.

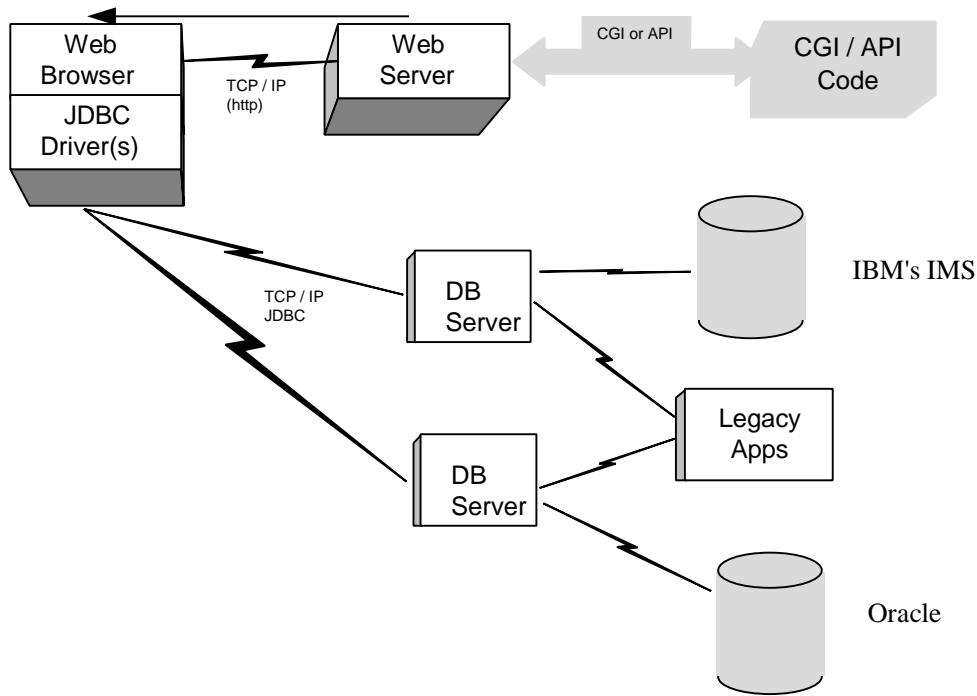
1.2.6.2. Proprietary Server

Database vendors in a first time were offering proprietary solution with sometimes a specific web server. If this solution still exists, this not a standard one and most of the main database vendors now are offering more open solution based on the two other solutions described here. These proprietary solutions should not be taken into account within the renaissance project.

1.2.6.3. Java / JDBC

JavaSoft has developed a standard database access interface, the JDBC API. This API provides Java programmers with a uniform interface to a wide range of relational databases, and provides a common base on which higher level tools and interfaces can be built. Leading database, connectivity, and tools vendors have already endorsed the JDBC API and are developing products using JDBC. JDBC drivers are currently available from a number of these vendors, providing Java connectivity to a wide variety of DBMS.

The JDBC API defines Java classes to represent database connections, SQL statements, result sets, database metadata, etc. It allows a Java programmer to issue SQL statements and process the results. JDBC is the primary API for database access in Java. **Note that JDBC can be used with any database for which a JDBC driver is available, this can including object-relational DBMS as well as non-relational DBMS such as IBM's IMS.**



The JDBC API is implemented via a driver manager that can support multiple drivers connected to different databases. JDBC drivers can either be entirely written in Java so that they can be downloaded as part of an applet, or they can be implemented using native methods to bridge to existing database access libraries.

Tools are available on the market to help integration of legacy database using this technology, they will be discussed in next chapters.

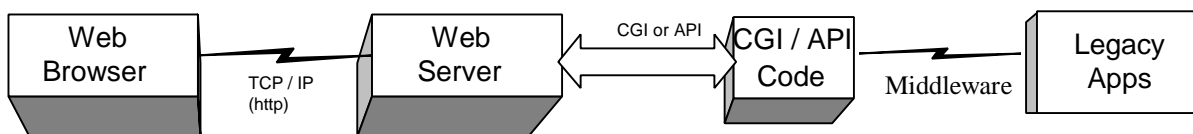
1.2.7. Linking Legacy Applications

The press has focused a lot of attention on Java, a language that can facilitate downloading applets on client machines that are equipped with Java interpreters. More visionary individuals have proposed that companies can maintain a single copy of a program like Microsoft Word and that each employee, when they want to use a text processor, will simply erase word from Ram by exiting the browser (it will provide many advantages: employees may have access only to the programs they need and they will always access the latest version because they download it from a central point which will manage update).

In a similar way, a company could arrange for employees to access existing legacy applications via browsers. In effect, an employee would access the corporate accounting application by going to the “ accounting Web site ”. They are mainly two means to develop such connection and of course they are quite similar to those use to connect to database.

1.2.7.1. CGI / API

Same principles as described above for database connection. Some tools use this approach to link CGI to legacy applications by directly offering language connection like Cobol, C and C++. However, you must have an API for your application which is available on the server platform. Some other tools provided gateway between usual Web server (Unix and NT) and legacy system (IBM mainly). It is so possible to connect quite easily a legacy application on a legacy system to a Web server.



1.2.7.2. Middleware connection

As I mentioned previously, Microsoft Explorer and Netscape Navigator already provide middleware connection through respectively Active X and Java + Corba (IIOP). This is a first version and it will be improved in the near future. It exists tools that already used this approach or plan to use it to connect to legacy application and database but most of the tools which are available now are using their own middleware components. However the tools which integrate the state of art at the moment are Netscape One from Netscape and Normandy from Microsoft. Normandy from Microsoft provides a proprietary solution whereas Netscape provides an open solution !

This new class of applications, called network-centric applications, combines the advantages of central control and administration with the power and flexibility of distributed applications.

In contrast to client-server applications that build static connections between a client and its server, network-centric applications accommodate collections of cooperating applications and network services. Applications are created by building new services (either client- or server-side) that transparently integrate and customize the services offered by intranets and the Internet. When requested by a user or program, the platform-independent client part of the application is downloaded from the application server and executes on the Netscape Navigator (or any other open-standards-compatible) client. The client then uses the open Internet protocols to interact with the server-based parts of the application as well as the standard services of the intranet and Internet.

2. Role in evolution process

(The part played by that technology in the general evolution process.)

The internet technologies will play an important role in the re-engineering/evolution process. Because it can be connected to old application as well as new one, it will not really impact the legacy application and so it can be a solution that can be rapidly provided and which can evolve when application evolve. Rad technology should be used to do such development.

In the early phase of the Web, applications were custom coded in either C++ or PERL, due to the lack of mature tools for building commercial applications. This approach is timely and costly, particularly in maintenance of applications once they are built. As the Web/database tools market continues to evolve, vendors are moving towards the model seen in client/server tools of providing customers with rapid application development environments that provide fast time to market without sacrificing flexibility in development.

In the client/server world, rapid application development (RAD) using a high level GUI tool significantly cuts time in application development and maintenance over custom coding. Many of the successful RAD client/server tools, such as PowerBuilder and Delphi, automate the process of client/server application development by providing wizards that generate application objects. These tools eliminate the bulk of programming from commercial applications.

Successful client/server tools have also been tightly integrated with underlying languages for business logic and application extensibility. This integration provides the benefits of rapid application development without sacrificing flexibility. Client/server tools that are based on languages with underlying object oriented class libraries provide a mechanism for reusing and extending the capabilities of the underlying language.

As Web/database tools mature, they are integrating GUI-based rapid application development with scripting languages to build in flexibility and to enable customers to get on the Web with commercial grade applications quickly.

However, when migrating a legacy application, it should be very important to take into account all the possible connection which will be done on this new application (classical GUI, or Web interface) to design it in a way that will be efficient for both solution.

Migrating from legacy systems should be done step by step ensuring the performance and reliability of the resulting system. Internet helps to make this possible by providing gateways between new and old software components mainly using middleware gateways and databases gateways.

3. Relevance to Renaissance

(The specific relevance of the technology to Renaissance and how it might be part of the Renaissance method.)

Internet technology can also be applied to internal application development and information dissemination, the so called Intranet. This allows companies to easily and cheaply process and deliver applications and information across their organisation with all the distribution problems of PC client/server development avoided. Business Week claim that “ the Intranet market will grow even more rapidly than the Internet market. They quote Zona Research’s forecast of Intranet growth from \$476 million in 1995 to \$8 billion by 1998”. That is why renaissance should take into account this technology. In France, we know the importance of minitel server and we can easily imagine what will be the demand and so the future of intranet and internet.

In the internet overview made below, I describe several software component, I summarize in the following table each with the interest they provide for renaissance.

Software components	Relevance for renaissance
Client Browser Software	Medium
Internet Search Engines	Low
Server Software	Low
Web Sites Utilities	Low
Web Site Development Software	High
Linking Company Databases	High
CGI, Server API	High
Proprietary Server	Low
Java / JDBC	High
Linking Corporate Applications	High
CGI, Server API	High
Middleware connection	High

Relevance of Internet Software component for Renaissance

3.1. Web vs. Client/Server: Rapid Application Deployment

The Web is attractive as an application platform because it eliminates the time and cost associated with application deployment. In the Web environment, deployment is instantaneous because the application resides on the server rather than the client. From the server, the application can be accessed from anywhere in the world. For system managers, this eliminates much of the headache involved in maintenance upgrades and in the administration involved in managing multiple platforms across multiple offices. For business managers, global access of server-side applications provide a compelling new channel for reaching new customers with products and services.

The Web also introduces a natural three-tier architecture that provides a foundation for scalability. With LAN-based client/server applications, there are typically a small number of simultaneous users. Applications have fat clients and users are logged onto the database from the start to the finish of a session. Organizations hit the client/server "wall" when the software demands of the application exceed the capability of the client, or when the number of database users can no longer scale. The Web, in contrast, supports a large number of simultaneous thin clients that access a server-side application. These users log onto the database for only the length of each transaction, rather than for the entire session.

4. Available support

(A discussion of available tool support with examples if appropriate.)

A lot of tools and language libraries are available for database connection, take a look at the following URL which identify and discuss most of them:

http://cscsun1.larc.nasa.gov/~beowulf/db/all_products.html . It identify product to access database but some of this tools also provided legacy application connection

For connection to legacy application, tools like Amazon, NetDynamics , Active Software, Java environment development including JOE, Tools using activeX and of course Netscape One from Netscape and Normandy from Microsoft should be used.

Within Cap Gemini Innovation we used some of them:

- {S,G,N}DBM Perl libraries to connect to {S,G,D}DBM database files. This is used in one of the renaissance web server for example.
- ColdFusion with its ODBC interface for a travel agency and for a food company within its intranet
- DBWeb from Microsoft for an internal web server.
- KBL a proprietary solution of Oracle for a web server for a museum
- IDQ HTX for our information highway implementation : <http://www.inforoute.cgs.fr>

The more powerful solution for integrating ODBC database on NT servers seems to be Cold Fusion. However for connecting legacy database or applications, we have for the moment no application.

5. Maturity assessment

(An assessment of the maturity of the technology.)

The maturity level is different for each software component identify previously in this document. The following table make a synthesis of their maturity

Software components	Maturity
Client Browser Software	Medium/High
Internet Search Engines	Medium/High
Server Software	Medium/High
Web Sites Utilities	Medium
Web Site Development Software	Medium
Linking Company Databases	From High to Medium
CGI, Server API	High
Proprietary Server	Medium
Java / JDBC	Medium
Linking Corporate Applications	Medium
CGI, Server API	Medium
Middleware connection	Medium

Maturity assessment of Internet Software components.

6. Inapplicability

(Areas where the technology should NOT be applied (e.g. 4GLs for real-time systems))

The Web, while providing significant benefits, introduces new technical challenges with respect to scalability, and management of session, state, transactions and security. These challenges are as follows:

- Web applications can face unpredictable and potentially enormously peak loads. This demands a high performance architecture that is extremely scalable.
- The Web is a stateless environment, in which the client and server are loosely coupled.
- Applications must keep state information from one page to another if they are to avoid requiring users to re-enter information such as user name and password from page to page.
- The Web presents new security issues for companies wishing to make internal databases accessible to external users. User authorization and authentication are more challenging in the Web environment because of the large number of potential users.

- The Web is changing rapidly and standards are still evolving.
- Internet components should be used carefully in systems that are demanding real-time responses, as heterogeneous platforms connected via middleware may slow down the system.

Until recently, developers have solved some of these issues through external mechanisms such as client-side "cookies" for managing state. More recently, development tools have begun to integrate built-in support for addressing these issues within a development environment.

7. Future development

(An assessment of possible future developments of the technology.)

Netscape and Microsoft future plan are described in the first part.

7.1. JDBC Future Plans

JavaSoft has plans for a higher-level API on top of JDBC, mapping Java objects transparently to rows of one or more relational tables. They call this a Java Object-Relational Mapping; the API should conform to the ODMG standard one listed above. JavaSoft is doing this work jointly with a partner; Their plans will be announced later this year. Their plans include a repository containing information about Java classes, including the database mapping information. They will work with partners on the definition of this repository to allow integration of development tools from many vendors.

JavaSoft also has plans for a Java Transaction Service API and implementation. This API will allow a Java program to commit a distributed transaction across multiple databases or other transaction-based resources. Again, JavaSoft is doing this work in co-operation with partners with expertise in transaction processing.

7.2. ActiveX

ActiveX Controls are interactive objects, created by programmers, that can be embedded in Web pages to enhance the experience of a Web site. For example, an ActiveX video control could be used to enhance a Web page with real-time video sequences.

The controls are language-independent, and can be programmed using programming languages such as C++, future versions of the Microsoft Visual Basic programming system, or Java in the future. Over 1,000 ActiveX Controls are available today from a wide variety of software vendors. Using the ActiveX Control Pad, Web developers can now easily incorporate ActiveX Controls into an HTML document using a simple point-and-click process.

8. Other comments

(Other comments regarding internet technologies).

Tools are now more and more powerful and are still maturing but some are really impressive ! To conclude, a comparison of the main technology usefull for Renaissance. This is the Netscape point of view and mine too.

8.1. Comparing Netscape One and Active X

Developers frequently ask how the Netscape ONE technologies compare to Microsoft's ActiveX initiative. activeX is basically a branding campaign to reposition Microsoft's 8-year-old OLE/OCX technology for the Internet. Because ActiveX originated as a desktop technology, however, it suffers from a number of problems. In contrast to Netscape ONE, which relies on open Internet standards and Java (both of which were specifically developed for the Internet), ActiveX controls are:

8.1.1. Platform- dependent

An ActiveX/OLE control is a piece of binary code that can run only on the platform for which it was compiled. To support the range of computers on the Internet, an ActiveX developer would have to recompile the control for every possible platform that might want to use the control, leading to much higher development and maintenance costs than an equivalent Java applet. In addition, as new computing architectures become available, every ActiveX/OLE control will have to be ported to the new platform. In contrast, Java is completely platform-independent, and Java components will run without recompilation on existing and future platforms that support Java.

8.1.2. Windows-centric

Microsoft claims that ActiveX/OLE will eventually work within Apple and Unix architectures, but it has not given any specific dates. It's worth noting that this effort has been subcontracted to an outside vendor to complete. Microsoft also indicated at the 1996 Professional Developer's Conference that ActiveX/OLE support on non-Windows platforms may not be free, nor are any other operating system vendors integrating ActiveX/OLE into their systems. In contrast, virtually all major computer and operating vendors (including Microsoft) have announced support for Java, making Java-written applications truly platform-independent and Internet-ready.

8.1.3. Insecure

ActiveX security is based on "code signing," a method for assuring that the ActiveX control has not been modified after it was signed. The problem with this is that code signing simply guarantees a vendor's identity; it says nothing about the code itself. Because ActiveX controls are binary code, once the control has been loaded, it can (either maliciously or unintentionally) damage users' information. In contrast, Java applets are designed from the ground up to be security-enhanced, and the Java "sandbox" security model checks the program before execution.

8.1.4. Proprietary

As a proprietary technology offered only by Microsoft, ActiveX has gained little industry support from OS vendors. In contrast, Java and the other Netscape ONE technologies have been licensed broadly, are available on all major computer platforms, and are supported by virtually every major computer vendor (including Microsoft).

9. References

[OOS] Object Oriented Strategies 'The monthly newsletter for manager & developers of object oriented systems', monthly issue from 1993 to 1996

[Netscape One] Netscape Document 'THE NETSCAPE ONE DEVELOPMENT ENVIRONMENT VISION AND PRODUCT ROADMAP' http://home.netscape.com/comprod/one/white_paper.html

[ActiveX] Microsoft documentation 'The Microsoft Active X Control Pad'