



KUNGL. TEKNISKA HÖGSKOLAN
Royal Institute of Technology

Evaluation of a Java-based Office-package

Tobias Öbrink

TRITA-IT R 97:21
ISSN 1103-534X
ISRN KTH/IT/R--97/21--SE

Department of Teleinformatics



KUNGL. TEKNISKA HÖGSKOLAN
Royal Institute of Technology

Evaluation of a Java-based Office-package

Tobias Öbrink

This is an evaluation of one of the first 100% Pure Java certified products, meaning that it should run acceptably on a range of Java-compatible computing platforms. The evaluation was done to see if the combination state-of-the-art Java technology and a Client-Server network model was ready for professional office use, considering usability, user-friendliness, performance, security and so forth. The Java Office package evaluated was not deemed good enough for use in a production environment yet but considering it took only about a year to develop, the result is still impressive. In the near future these kinds of applications will probably be a common sight in offices and a part of the application flora offered in set-top boxes, network computers, netPCs and other consumer devices

TRITA-IT R 97:21
ISSN 1103-534X
ISRN KTH/IT/R--97/21--SE



Department of Teleinformatics
Telecommunication Systems Laboratory

1. Introduction	1
1.1 What this report covers	1
1.2 Outline of this Document	2
2. The Anyware Java Office	3
2.1 How it works	3
2.1.1 Logging in.....	3
2.1.2 The Anyware Java Display Client	4
2.1.3 The Anyware Office	5
2.2 The Builder	6
3. Test-site set up	7
3.1 Server platform.....	7
3.2 Client platforms.....	7
4. Timing.....	9
4.1 Loading and starting up.....	9
4.2 Interaction	11
5. Features	13
5.1 Server side bug	13
5.2 General for all tested configurations	13
5.2.1 General for all applications.....	13
5.2.2 Words.....	13
5.3 Features on platform 1	14
5.3.1 Words.....	14
5.3.2 Spreadsheets.....	14
5.4 Features on platform 2	14
5.4.1 Words.....	15
5.5 Features on platform 3	15
5.5.1 Crashes.....	15
5.5.2 Words.....	15
5.6 Features on platform 4	15
5.6.1 Words.....	16
5.7 Features on platform 5	16
5.7.1 Restart error?.....	16
5.7.2 Debugging.....	16
5.7.3 Words.....	16
5.8 Features on platform 6	16
5.8.1 General.....	16
5.8.2 Exiting.....	16
5.9 Features on platform 7	17
6. Security.....	19
6.1 User Authentication	19
6.2 File Access	19
7. Conclusion.....	21
8. References.....	23

1. Introduction

What is a Java-based Office package? By office package I mean a suite of applications for office use in a commercial enterprise. This suite of applications typically include a word processor, a spread sheet, an email handling tool and some support for interfacing to database- and real-time services such as Rueter's Triarch[6]. Such tools have been available for quite a while now and most people have an idea of what these tools should offer in terms of functionality.

By Java-based I mean that the tools are written in the Java computer programming language, at least partly. The Java programming language[1] is an object-oriented language that was designed to be easy to learn and get started. The source code is compiled into a byte-code that is designed to be easily and compactly transferred over most networks. This byte code can either be interpreted by a virtual machine[2] that translates the virtual machine instructions into platform dependent code, or it can be compiled at the destination machine before being run. None of these ideas are new inventions, so Java is rather a carefully selected mix of successful software engineering concepts.

What all this means in practice is that the suite of Office applications can be run on any platform equipped with a Java virtual machine, i.e. any platform with a Java-enabled web browser installed.

What was most interesting in this evaluation was to see if the combination state-of-the-art Java technology and a Client-Server network model was ready for these kinds of services, considering usability, user-friendliness, performance, security and so forth.

1.1 What this report covers

The evaluation performed was mainly based on the following reasoning:

- Due to the fact that the Java byte code generally is interpreted by a virtual machine which is a part of- or started by a web browser we can guess that the client-part of the applications will be quite slow. The question is how slow, and which minimal configuration is needed to get sufficient performance for acceptable working conditions.
- Due to security reasons, the virtual machine in web browsers are not allowed to access the local machine's file system. This means that all files generated by word processors and spreadsheet programs and the like must reside on the server from where the Java byte code originated. Therefore Java-based applications designed to run from a WWW page is inherently Client-Server oriented and sensitive to server response times. Scalability and computation-distribution should be other important issues to study, but is not included because of lack of time and manpower.
- For users to want to use the system a minimum level of security must be supported. Such as basic authentication and file access handling.
- Last but not least, the Java language was at the time quite new and not very strictly standardized so far, so robustness on different platforms is an issue that is covered.

The evaluation consisted of collecting, analysing and presenting timing data and debugging information from running a Java-based Office package called Applix Anyware(see section 2), and drawing conclusions.

1.2 Outline of this Document

In section 2, the Java-based Office package evaluated is presented. Section 3 describes the test-site set up. In section 4 the collected timing information is presented together with some discussion. Section 5 presents a report on unwanted features and bugs in the evaluated application suite. In section 6 discuss the minimal security configuration supported by the evaluated Java-based Office package. The concluding discussion is found in section 7.

2. The Anyware Java Office

The Applix *Anyware* Java Office™[3] tool suite is a product of Applix Inc. and is built from a Unix-based Office suite called *Applixware*™. The functionality is approximately the same as in Microsoft Office™ or Corel® Office. Anyware Office was one of the first products certified by Sun Microsystems as a 100% Pure Java™[4] product. Sun Microsystems' 100% Pure Java Program assures users that an application certified as 100% Pure Java will run acceptably on a range of Java-compatible computing platforms.

Applix Anyware uses a Web entry page, but the Anyware application server can be located on a different machine than the Web server. This is the recommended configuration, but for the evaluation I had an otherwise not used Web server at my disposal so I deemed this may not be needed. When I later upgraded the Web server to a Netra J with attached JavaStation, it ran out of process memory quite fast when using Words, though.

The AxMain program, that is started when a client connects, uses some 40 MBytes memory at startup and increases quite fast when using Words. So it's a good idea to make sure you have enough, preferably more than 100 MBytes, available swap space on the server machine.

Applix Anyware also needs an X server to be running on the server machine. If not, the client exits with the message "missing display". The server also needs the X access control to be turned off. If not, the client exits with the message "The server has exited", which may not be the most informative error message. The reason for all this is explained in section 2.1.2 below.

The client machine should have a Web browser supporting Java 1.02 or later.

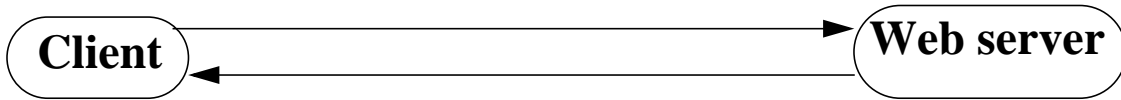
2.1 How it works

The Anyware solution is based on the concept that the user interface should be where the user is, and the computing should happen where the computing power and data are. Thus, relying on Java to do the lighter user interface tasks, but not the more intensive calculations and processing for which its interpreted nature is much less suited. Figure 1 gives an overview of the Start-up procedure, Figure 2 shows how subsequent user interaction is handled and Figure 3 shows a screenshot of the Java Office applet window.

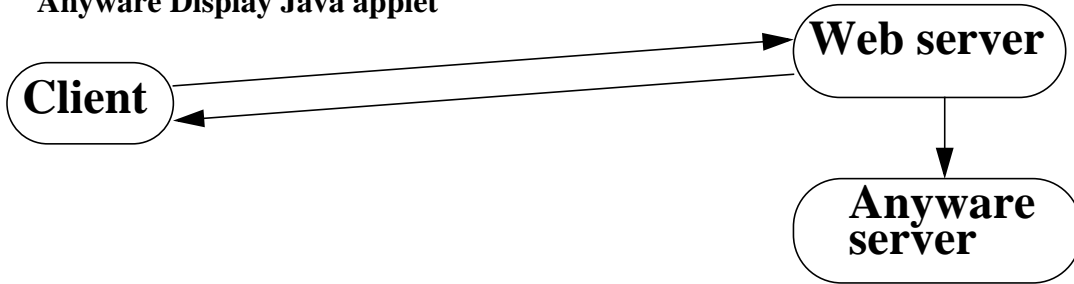
2.1.1 Logging in

When a user logs into a Web Server that is running an Anyware frontend, that user is authenticated by a username/password combination by the Web server. The Anyware application server in itself has no authentication procedure. See Figure 1 for an overview of the start-up procedure.

1 Client requests applet-page and optionally performs authentication



2 Web server starts the anyware server and the Client loads the Anyware Display Java applet



3 The Anyware Display Java applet loads the Anyware Office Java applet

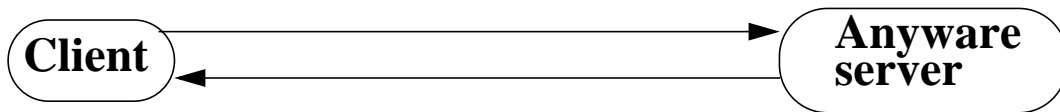


Figure 1. Overview of the start-up procedure

2.1.2 The Anyware Java Display Client

Once a correct username and password has been entered the Anyware Java Display Client, an approximately 570 KB Java applet is downloaded to the client machine. This applet is the “white board” on which any application’s interface created using the Anyware tool set can be “drawn”. The Anyware display client contains Java widgets and controls such as buttons, menus and list boxes. Relying on the Java client widgets to create the user interface enables the interface to mimic the look of the native applications of the operating system on which it is being run. So, if the client is on a Windows 95 desktop the application will have the look of the standard Windows widget set, but if the client is running on a UNIX desktop, the application will have the look of the Motif widget set. Figure 2 shows how the Anyware Java Display Client works.

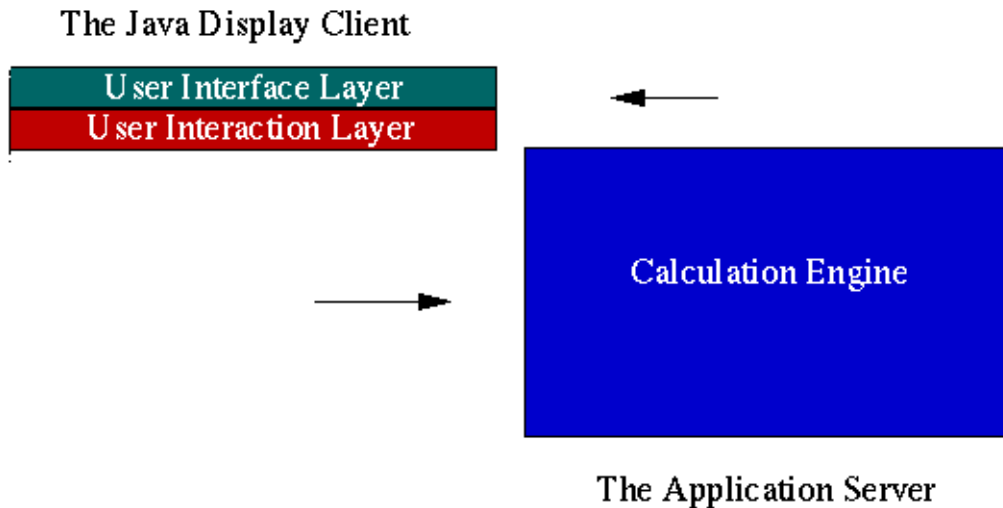


Figure 2. User interaction handling in the Java Display Client

2.1.3 The Anyware Office

The Web Server passes the communication channel to the Application Server which starts up the default application, the Anyware Office Java applet. The Anyware Office Java applet window looks like Figure 2.



Figure 3. The Anyware Office Java applet

In this window you can click on the icons to start the different applications or you can use the menus. Then the appropriate engine is started on the application server, while the description of what the application interface looks like is downloaded to the Java Display Client. The Java Display Client then builds the interface and displays it to the client. So the only information that is being sent from the application server over the network is the description of what the application should look like. And, the only information being sent from the client to the application server is user interaction information.

2.2 The Builder

Custom applications can be created using the Builder module of the Innovator's Workbench. Builder allows developers to use the Applix ELF language or C++ modules to create applications. Builder includes 4100 pre-defined macros for rapid application development. Builder also supplies a graphical interface design tool, debugger, data access tool and a Class Browser.

3. Test-site set up

The tests were run during normal network conditions on the department network. The following Client-Server configurations were tested.

3.1 Server platform

A SUN Netra j with the Solaris 2.5.1 operating system [5]. The Web server used was a Netscape Enterprise server. Both the Web server and the Anyware application server ran on the same machine, although it is recommended to locate the Anyware application server on a separate machine on the same subnet as the Web server.

3.2 Client platforms

The following client platforms and network connections were evaluated (see Table 1).

Table 1: Client platforms

Client platform number	Web browser	Machine	Operating system	Network connection
1	Netscape 3.01 Gold	SUN SPARC Station 10	Solaris 2.5.1	ISDN Duo 128kbps
2	Netscape 3.01 Gold	SGI Indy	IRIX 5.3	Switched ethernet 5-8 Mbps
3	Netscape 3.01 Gold	SUN SPARC Station 10	Solaris 2.4	Switched ethernet 5-8 Mbps
4	Netscape 3.01 Gold	HP Vectra 562	Windows 95	Switched ethernet 5-8 Mbps
5	HotJava 1.0 LAR	SUN JavaStation 1	JavaOS 1.0 LAR	Switched ethernet 5-8 Mbps
6	Netscape 4.0 b5	SGI Indy	IRIX 5.3	Switched ethernet 5-8 Mbps
7	Netscape 3.01 Gold	SUN Netra J	Solaris 2.5.1	Loopback

Platform 1. Netscape 3.01 Gold, running on a SUN SPARC Station 10 with the Solaris 2.5.1 operating system. The client machine was connected to the server machine via an Ethernet to an Ascend P75, over ISDN Duo 128kbps to an Ascend Pipeline and over Switched Ethernet the last part. 512kbps maximum throughput with data compression. 128kbps without.

Platform 2. Netscape 3.01 Gold, running on a SGI Indy with the IRIX 5.3 operating system. The client machine was connected to the server machine via switched ethernet which gives ca 5-8 Mbps.

Platform 3. Netscape 3.01 Gold, running on a SUN SPARC Station 10 with the Solaris 2.4 operating system. The client machine was connected to the server machine via switched ethernet which gives ca 5-8 Mbps.

Platform 4. Netscape 3.01 Gold, running on a HP Vectra 562 with the Windows 95 operating system. It uses a Just-In-Time compiler to speed up execution of Java applets. The client machine was connected to the server machine via a Switched Ethernet which gives ca 5-8 Mbps.

Platform 5. HotJava 1.0 LAR beta on a SUN JavaStation 1 with JavaOS 1.0 LAR alpha operating system [5]. The client machine was connected to the server machine via a Switched Ethernet which gives ca 5-8 Mbps.

Platform 6. Netscape 4.0 b5, running on a SGI Indy with the IRIX 5.3 operating system. The client machine was connected to the server machine via switched ethernet which gives ca 5-8 Mbps.

Platform 7. Netscape 3.01 Gold running on a SUN Netra J with the Solaris 2.5.1 operating system. The Client was on the same machine as the Server using only loopback communication. No network involved.

4. Timing

The collected timing information consists of both measured times and estimated times. Times were measured for actions that was expected to take longer times such as loading, starting up applications and displaying windows. For actions that was expected to take too short time to be reliably measured, such as feedback times for interaction, the times were estimated. The fault variances for measured times is about 1 second while the fault variances for estimated times is about 0.5 second.

The factors contributing to overall delay is network performance, server load, clients CPU performance, clients operating system performance, other processing load on the client platform and Java runtime implementation (virtual machine or Just-in-time compiler).

4.1 Loading and starting up

When looking at the timing data collected below (Table 2) it seems network latencies is not the major factor slowing down the application when loading and starting up the different applets. Except from loading the initial Anyware applet (whose size was more than 570KBytes) the timing information gathered seems not to differ in proportion to difference in roundtrip time. Platform 1 was connected via ISDN and switched ethernet while platforms 2 - 6 was connected via switched ethernet and finally platform 7 was run on the server machine. When looking at the graph in Figure 4, that is generated from the data in Table 2, one can easily see the performance on the different client platforms.

Table 2: Applet timing on different platforms

Applet Timing	1	2	3	4	5	6	7
Load Anyware Applet	55 sec	13 sec	33 sec	17 sec	27 sec	17 sec	3 sec
Java Office	23 sec	19 sec	20 sec	14 sec	5 sec	21 sec	14 sec
Words	22 sec	23 sec	28 sec	18 sec	16 sec	25 sec	8 sec
Spreadsheets	16 sec	21 sec	20 sec	13 sec	16 sec	19 sec	5 sec
Mail	10 sec	15 sec	14 sec	5 sec	4 sec	9 sec	4 sec
HTML Author	12 sec	20 sec	16 sec	10 sec	5 sec	15 sec	6 sec
Directory Displayer	5 sec	8 sec	5 sec	3 sec	2 sec	5 sec	2 sec
Data	8 sec	15 sec	16 sec	5 sec	3 sec	8 sec	5 sec
Macro Editor	8 sec	8 sec	14 sec	7 sec	4 sec	10 sec	4 sec
Graph. intense popup	19 sec	13 sec	15 sec	16 sec	4 sec	13 sec	5 sec
“Normal” popup	4 sec	7 sec	6 sec	2 sec	2 sec	5 sec	2 sec

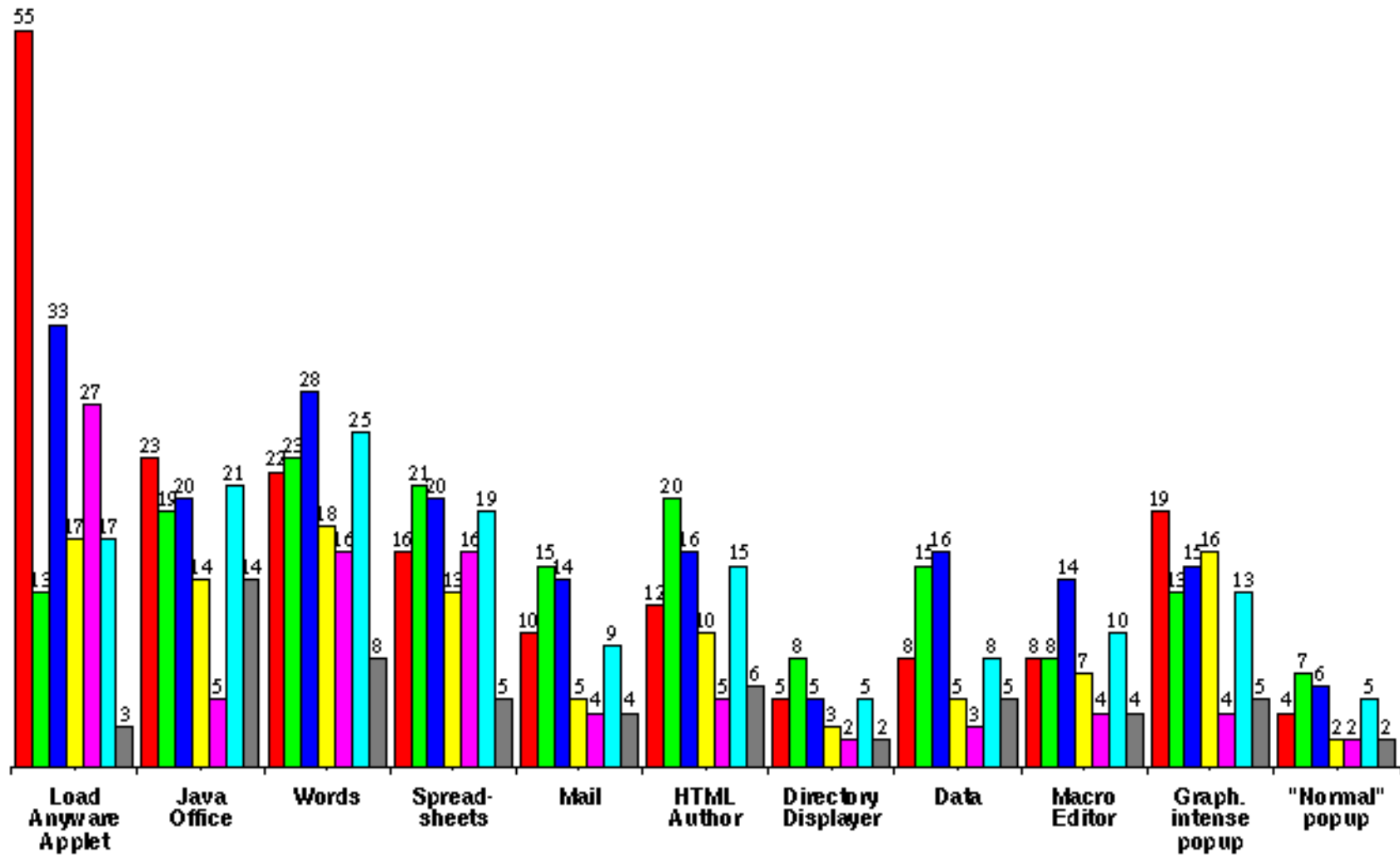


Figure 4. Applet timing on different platforms

The “Load Anywhere Applet” is the time from that the page is loaded until the “100% loaded” message (see section 2.3.2). The “Java Office” is the time from that the loading is finished until the Java Office window is fully displayed. The rest of the fields is the time from mouse click until the window is fully displayed and further interaction is possible. The times given are averages from three different test runs on each of the platforms. Test data can be found in Appendix A.

The max bandwidth difference between ISDN and switched ethernet when it comes to loading the different applets seems to be insignificant. Platform 1 and platform 3 is the same type of machines only differing in operating system versions and network connectivity.

From the tests on platform 7, where no network was involved, we can see that the network does include some delay. The figures are somewhat dependant on the faster processor in platform 7 as well so the difference is not completely due to absence of network delay. Therefore the network delay was measured separately (Table 3).

Table 3: Network roundtrip times to test platforms

1	2	3	4	5	6	7
32 ms	2 ms	1 ms	2 ms	< 1 ms	2 ms	none

The network roundtrip times were measured by using the program “ping” from the server machine.

Platform 2 and 6 are the same machine, but using different browser versions. Apparently no significant improvement in performance of the Java virtual machine was implemented.

From the tests on platform 4 we see that a Just-In-Time compiler gives a noticeable speedup, although in this case as in platform 7 the faster processor may affect the results.

On platform 5 a JavaStation was used, optimized for running Java applets with minimal overhead. In this case we can see a turning point where the network conditions and server load may affect the user experience significantly. The JavaStation resided on the same subnet as the server machine resulting in very low roundtrip times.

4.2 Interaction

It was expected that the feedback from interaction(response time) should be somewhat proportionate to the bandwidth available, but on those connection types that was investigated this proportion turned out not to be the most significant factor. A setback for the testing was the fact that the Ascend ISDN router used compression on the transmitted data which gives a somewhat stochastic behaviour of the application. The router also disconnected after a timeout if no data was sent, potentially resulting in unexpected conditions for the application.

The response times on Platform 5, the JavaStation, was considerably lower than on the other platforms even if it had the same connectivity as most of the other platforms. However, even with those reduced response times the interaction was slow. Typical response times for inserting a character ranged from tenths of seconds up to 3 seconds. It depends on what kind of charater you insert - if you insert a newline just before a page end in the beginning of a 6-page paper it will take a lot longer time.

Deleting characters was a little bit slower. Pulling down menus was much slower. Repositioning the cursor with mouse clicks was very fast most of the time as was pushing buttons. Moving sliders was very slow sometimes and resizing windows was painfully slow, for reasons described in section 5.

5. Features

Or bugs, or (un)wanted functionality or what you want to call it. First is presented some general behaviour on all platforms and after that follows some platform-specific behaviour. The evaluated application suite seems to be fairly platform independent since most of the “bugs” appeared on all the tested configurations. I’ve tried to keep the descriptions brief and moved details of the bug reports, such as traces and debugging information, to Appendix B.

5.1 Server side bug

The server side of the application seems to use a lot of memory, since a “Memory Limit exceeded” warning pops up sooner or later in a window on the server machine. I consider this a bug since no information is forwarded to the user. The client just locks and the user don’t know why, or what he/she can do about it. The server administrator in turn will be greeted by a message suggesting he should free some memory by closing windows and behind that message a blank pop-up window launched by the Anyware application server, AxMain.

5.2 General for all tested configurations

The following appeared on all configurations tested.

5.2.1 General for all applications

NullPointerException. This exception was often encountered after a series of mouse clicking in different parts of the screen. Also routinely encountered when working in a popup-dialogue-series such as the Spreadsheets Chart drawing wizard and the Words Style-dialogue

Cut and Paste. Within the Anyware applications it works fine. From any other part of the screen not.

X refresh. Does not issue a refresh request to the Anyware application server.

Delays. Sometimes the applications hang for minutes, maybe waiting for a response from the application server or maybe something else. No reason was given.

Resizing window. Resizing a window results in a long waiting while the application re-drawing itself for every resize-event it receives.

Window-sizes. The default window sizes of the applications were too small. Since resizing a window results in a long waiting while the application re-drawing itself for every resize-event it receives, it is important to have fair default window sizes.

Menus. Some pulldown menus stay after one click, some others require a drag.

Exit. Sometimes it takes minutes for applications to exit for no obvious reason.

5.2.2 Words

Bugs and missing functionality specific for the Words applet.

Drag-and-select. The slider doesn’t follow the mouse pointer when doing a drag-and-select outside the windowframe.

Editing “forwards”. The last character typed is not echoed until you do some more input.

Editing “backwards”. When inserting text in front of a string, making the string to pass a page boundary, the string disappears and you have to edit or step through the string to make it visible again.

Before the string gets visible again you cannot delete a preceding newline. You have to press at least two or three times to delete it.

If you ignore that the following string disappeared and just continue to type and then insert another newline before the string, it becomes visible again, but some other part may become invisible instead.

Select->Delete. Deletion of too much text within a short time makes the Words window die and an error popup window appears saying “memory error, incorrect free'er, pid=9 p->pid=513 Aborted”. If you start up Words again and repeat the same behaviour, then not only will Words die but also the AxMain program on the server machine, resulting in that the AnywareOffice applet exits with a “Anyware server has exited”-errormessage. There is a way to avoid this - by only select->deleting a little at a time and wait for a while between each, and above all press Save between each selection->deletion.

File->Page Setup. Defaults to US Letter and PCL5 printer even if I previously defined the document to have other proportions and Postscript as the preferred printer type.

Stepping through text with arrow keys. Holding down an arrowkey for a while will result in minutes of waiting while the cursor steps one character per second. Hacking ferociously on the same arrowkey will result in the cursor speeding quite fast.

5.3 Features on platform 1

The platform 1 consisted of Netscape 3.01 Gold, running on a SUN SPARC Station 10 with the Solaris 2.5.1 operating system. The machine was connected to the Anyware application server via an Ethernet to an Ascend P75, over ISDN Duo 128kbps to an Ascend Pipeline and over Switched Ethernet the last part. The maximum throughput with data compression is 512kbps. 128kbps without data compression.

5.3.1 Words

Stepping with arrowkeys. Works for a while, but then stops working for some reason.

Crashes. Insert page number -> delete page number -> resize window crashes Words.

5.3.2 Spreadsheets

Menus. The second-level arrowmenus doesn't popup when dragging the menu item. Only when clicking the menu open, and then entering the menu item.

5.4 Features on platform 2

The platform 2 consisted of Netscape 3.01 Gold, running on a SGI Indy with the IRIX 5.3 operating system. The machine was connected to the Anyware application server via switched ethernet which gives ca 5-8 Mbps.

5.4.1 Words

Style. Rightmost column. The list following “Create Style by:..” overlapped the buttons “Create Style” and “<something I cannot read> Style”

Stepping through text with arrow keys . When you step downwards and out of the window the first time, the slider doesn't follow the cursor. If you find the cursor again, using the slider, and then step upwards then the slider will follow the cursor.

When you edit the text resulting in creating of a new page outside the window, the slider doesn't follow the cursor.

If, when the slider and cursor is “out of synch”, you step past the end of the document, the applet fails and you can no longer edit the text. It was easily repaired by exiting Words and restarting it from the Anyware Office applet.

5.5 Features on platform 3

The platform 3 consisted of Netscape 3.01 Gold running on a SUN SPARC Station 10 with the Solaris 2.4 operating system. The machine was connected to the Anyware application server via switched ethernet which gives ca 5-8 Mbps.

5.5.1 Crashes

There was a few crashes that I suspect may have been caused by either the Netscape browser or -server rather than the Anyware application server.

On loading HTML page(AAlogin.ax). •Fails to load the Anyware Java Office applet.

On starting Words. jFails to start Text window.

5.5.2 Words

Stepping with arrowkeys. Works for a while, but then stops working for some reason.

After resizing the window. When insertion of text makes the cursor pass beyond the “original” window bounds, the slider moves one “current” pagelength down and all insertion is inhibited. After moving the slider upwards again, insertion of text is again allowed.

When insertion point have passed the “original” window boundary, the slider doesn't follow the cursor so that it is possible to edit the text below the windows edge.

Select->backspace: When trying to delete a few lines by selecting them and pressing backspace, no more editing can be performed.

5.6 Features on platform 4

The platform 4 consisted of Netscape 3.01 Gold, running on a HP Vectra 562 with the Windows 95 operating system. The machine was connected to the Anyware application server via a Switched Ethernet which gives ca 5-8 Mbps.

5.6.1 Words

Select-backspace. As noted under General for all platforms above, Select->backspace->select->backspace makes the Words window die and an error popup window appears saying “memory error, incorrect free'er, pid=9 p->pid=513 Aborted”. On platform 4, if you restart Words and repeat the same behaviour a second time the application freezes.

5.7 Features on platform 5

The platform 5 consisted of HotJava 1.0 LAR on a SUN JavaStation 1 with JavaOS 1.0 LAR operating system. The machine was connected on the same subnet as the Anyware application server via a Switched Ethernet which gives ca 5-8 Mbps.

5.7.1 Restart error?

Restarting the AnywareOffice applet after a crash by doing View->Clear Cache followed by View->Reload Page results in an error message window without content popping up. The AnywareOffice applet is re-loaded and restarted, however.

5.7.2 Debugging

There's no Java Console in the HotJava browser so no error stacks is available from the tests on the JavaStation.

5.7.3 Words

Resizing window to full screen. Be sure to have repositioned the bottom slider as far left as you can before “blowing up” the window, otherwise you will have a big window with only the right part of a page displayed and no way to “scroll in” the page into focus again.

5.8 Features on platform 6

The platform 6 consisted of Netscape 4.0 b5, running on a SGI Indy with the IRIX 5.3 operating system. The machine was connected to the Anyware application server via switched ethernet which gives ca 5-8 Mbps.

5.8.1 General

Netscape 4.0 b has the same java virtual machine as 3.01 Gold. It seems a little bit slower than 3.01 Gold, but maybe that's just beta-mania.

5.8.2 Exiting

After Anyware Office have exited or crashed, the Netscape window consider itself exited as well and doesn't respond to any interaction. You just have to kill the process from a shell. This may have been a bug in Communicator and not in Anyware, but noone else running applets on this beta version had experienced this.

5.9 Features on platform 7

The platform 7 consisted of Netscape 3.01 Gold running on the Web/Anyware application server, which is a SUN Netra J with the Solaris 2.5.1 operating system. No network was involved.

Because of lack of time no extensive testing could be done on this platform.

6. Security

Investigating security was not the prime objective of this evaluation since the programs were just released and security considerations seems to enter first when a product is stable enough to start marketing.

6.1 User Authentication

When logging in, Anyware used the Netscape Enterprise Web server's authentication for access to the file area containing the Anyware frontend. This means the Anyware application server has no authentication of it's own.

6.2 File Access

Included in the Java Office application suite is a Directory Displayer with which one can navigate in the underlying file system and setting access permissions and editing files and directories. In this case it relies on the underlying file system for handling the access rights.

7. Conclusion

Overall impression is unsurprisingly - slow and buggy. However, as soon as you've got past that the evaluated Java Office package worked surprisingly well. It also worked reasonably platform independent. All you need is a Java-enabled browser, some patience and to remember to save frequently, because it can be a little shaky so far. The first versions of this report was written using the word processor and spreadsheet in the evaluated Java Office package but it craved a lot of patience and stubbornness.

From the tests made it seems network latencies is not the major factor slowing down the application but rather the Java Virtual Machine. This in spite of the fact that almost all the work is done on the server by compiled native code.

In the evaluated Java Office package, network delays and server load combined did have some negative influence on interaction-intensive work, such as text editing. For example, the Java-based word processor interface had to communicate with the server for each and every inserted character or step with an arrow key. This architecture doesn't scale well in systems with many users spread over large distances or having bad connectivity.

The evaluated Java Office package was not considered good enough for use in a production environment but considering it took only about a year to develop, the result is still impressive. In the near future these kinds of applications will probably be a common sight in offices and a part of the application flora offered in set-top boxes and network computers.

From a system administration point of view the server-centric nature of the evaluated Java Office package allows to centrally manage the applications and information access, reducing the operating costs by eliminating the need to distribute, install and maintain applications software and/or data across a large number of desktops.

Existing distributed file systems support the central file system management part, but you still have to install and manage applications software for all different platforms in the system. The possibility to use a single runnable program for all the different platforms in a complex system like the one here at the Department of Teleinformatics or KTH overall is intriguing.

8. References

- [1] James Gosling, Bill Joy, and Guy Steele, "The Java Language Specification", Addison-Wesley 1997, ISBN 0-201-63451-1
- [2] Tim Lindholm and Frank Yellin, "The Java Virtual Machine Specification", Addison-Wesley 1997, ISBN 0-201-63452-X
- [3] More information on Applix Anyware can be found on the URL <http://www.applix.com/anyware/> or by contacting their sales representative in Sweden, Agenda AB, Box 19 533, S-104 32 Stockholm, Sweden, Phone: Int. +46 8 15 50 25, Fax: Int. +46 8 15 50 75, URL <http://www.agenda.se>.
- [4] 100% Pure Java Cookbook for Developers, --a technical document which defines what is and isn't 100% Pure Java and provides developers with rules and hints for maximizing the portability of Java programs. Can be downloaded from the URL <http://www.javasoft.com/100percent/cookbook.intro.html>
- [5] At the dept. of Teleinformatics, KTH we have a small but slowly growing Sun Technical Research an Excellence Center (SunTREC) that among other things include a Java Computing testbed. More information on the SunTREC can be found at URL <http://suntrec.it.kth.se/> and more information on Sun's Java computing strategy including Netra j servers and JavaStations can be found on the URL <http://www.sun.com/javastation/>
- [6] Triarch, from Reuters, is the world's most widely used digital financial information-management system. More information can be found on the URL <http://www.ims.reuters.com/triarch/triarch.htm>

Appendix A

Timing data from tests on different platforms.

Table 4: Sun SPARC 10 with Solaris 2.5.1 and Netscape 3.01 Gold over ISDN

Applet Timing	Test 1	Test 2	Test 3	Mean	Rounded
Load Anyware Applet	60	50	56	55.333333333	55 seconds
Start Anyware Java Office	30	18	20	22.666666667	23 seconds
Start Words	35	21	9	21.666666667	22 seconds
Start Spreadsheets	20	13	15	16	16 seconds
Start Mail	10	9	11	10	10 seconds
Start HTML Author	15	11	9	11.666666667	12 seconds
Start Directory Displayer	5	5	4	4.666666667	5 seconds
Start Data	10	5	8	7.666666667	8 seconds
Start Macro Editor	10	7	7	8	8 seconds
Open graphically intense popup	30	14	13	19	19 seconds
Open "normal" popup	2	4	5	3.666666667	4 seconds

Table 5: SGI Indy with IRIX 5.3 and Netscape 3.01 Gold over Ethernet

Applet Timing	Test 1	Test 2	Test 3	Mean	Rounded
Load Anyware Applet	10	15	13	12.666666667	13 seconds
Start Anyware Java Office	20	20	17	19	19 seconds
Start Words	25	23	22	23.333333333	23 seconds
Start Spreadsheets	24	20	20	21.333333333	21 seconds
Start Mail	15	15	14	14.666666667	15 seconds
Start HTML Author	22	17	20	19.666666667	20 seconds
Start Directory Displayer	10	7	7	8	8 seconds
Start Data	12	18	15	15	15 seconds
Start Macro Editor	10	8	7	8.333333333	8 seconds
Open graphically intense popup	15	12	12	13	13 seconds
Open "normal" popup	4	9	8	7	7 seconds

Table 6: Sun SPARC 10 with Solaris 2.4 and Netscape 3.01 Gold over Ethernet

Applet Timing	Test 1	Test 2	Test 3	Mean	Rounded
Load Anyware Applet	40	28	30	32.666666667	33 seconds
Start Anyware Java Office	25	22	12	19.666666667	20 seconds
Start Words	25	30	30	28.333333333	28 seconds
Start Spreadsheets	20	19	20	19.666666667	20 seconds
Start Mail	10	15	17	14	14 seconds
Start HTML Author	15	14	20	16.333333333	16 seconds
Start Directory Displayer	5	5	5	5	5 seconds
Start Data	15	15	17	15.666666667	16 seconds
Start Macro Editor	15	12	15	14	14 seconds
Open graphically intense popup	15	14	15	14.666666667	15 seconds
Open "normal" popup	5	6	6	5.666666667	6 seconds

Table 7: HP Vectra with Windows 95 and Netscape 3.0 over Ethernet

Applet Timing	Test 1	Test 2	Test 3	Mean	Rounded
Load Anyware Applet	20	15	15	16.666666667	17 seconds
Start Anyware Java Office	10	15	15	13.333333333	14 seconds
Start Words	15	25	15	18.333333333	18 seconds
Start Spreadsheets	8	20	10	12.666666667	13 seconds
Start Mail	7	5	4	5.333333333	5 seconds
Start HTML Author	9	12	10	10.333333333	10 seconds
Start Directory Displayer	2	4	3	3	3 seconds
Start Data	5	5	6	5.333333333	5 seconds
Start Macro Editor	10	5	6	7	7 seconds
Open graphically intense popup	15	19	14	16	16 seconds
Open "normal" popup	2	2	2	2	2 seconds

Table 8: Sun JavaStation1 with JavaOS 1.0 LAR and HotJava 1.0 LAR over Ethernet

Applet Timing	Test 1	Test 2	Test 3	Mean	Rounded
Load Anyware Applet	22	27	31	26.666666667	27 seconds
Start Anyware Java Office	3	8	3	4.666666667	5 seconds
Start Words	15	16	17	16	16 seconds
Start Spreadsheets	18	15	15	16	16 seconds
Start Mail	4	4	4	4	4 seconds
Start HTML Author	6	5	5	5.333333333	5 seconds
Start Directory Displayer	3	2	1	2	2 seconds
Start Data	3	4	3	3.333333333	3 seconds
Start Macro Editor	4	4	4	4	4 seconds
Open graphically intense popup	4	5	4	4.333333333	4 seconds
Open "normal" popup	2	3	2	2.333333333	2 seconds

Table 9: SGI Indy with IRIX 5.3 and Netscape 4.0b5 over Ethernet

Applet Timing	Test 1	Test 2	Test 3	Mean	Rounded
Load Anyware Applet	20	20	11	17	17 seconds
Start Anyware Java Office	20	18	25	21	21 seconds
Start Words	25	25	24	24.666666667	25 seconds
Start Spreadsheets	20	17	19	18.666666667	19 seconds
Start Mail	10	8	10	9.333333333	9 seconds
Start HTML Author	13	16	17	15.333333333	15 seconds
Start Directory Displayer	5	5	5	5	5 seconds
Start Data	8	10	7	8.333333333	8 seconds
Start Macro Editor	10	10	11	10.333333333	10 seconds
Open graphically intense popup	13	14	13	13.333333333	13 seconds
Open "normal" popup	7	3	5	5	5 seconds

Table 10: Sun Netra J with Solaris 2.5.1 and Netscape 3.01 Gold (server localhost)

Applet Timing	Test 1	Test 2	Test 3	Mean	Rounded
Load Anyware Applet	2	2	4	2.666666667	3 seconds
Start Anyware Java Office	18	10	13	13.666666667	14 seconds
Start Words	8	8	9	8.333333333	8 seconds
Start Spreadsheets	6	4	5	5	5 seconds
Start Mail	4	4	4	4	4 seconds
Start HTML Author	6	5	6	5.666666667	6 seconds
Start Directory Displayer	2	1	2	1.666666667	2 seconds
Start Data	5	5	5	5	5 seconds
Start Macro Editor	4	4	3	3.666666667	4 seconds
Open graphically intense popup	5	5	4	4.666666667	5 seconds
Open "normal" popup	2	2	2	2	2 seconds

Appendix B

This appendix contains some traces and debugging information obtained from the Netscape Navigator's Java Console upon encountering a bug.

General for all applications

NullPointerException.

The following exception was often encountered after a series of mouse clicking in different parts of the screen. Also routinely encountered when working in a popup-dialogue-series such as the Spreadsheets Chart drawing wizard and the Words Style-dialogue

```
java.lang.NullPointerException
  at java.awt.Rectangle.intersects(Rectangle.java:154)
  at page_t_Draw(page_t.java)
  at WPViewer.Q95(WPViewer.java)
  at WPViewer.V22(WPViewer.java)
  at WPViewer.B00(WPViewer.java)
  at AxViewer.run(AxViewer.java)
  at java.lang.Thread.run(Thread.java:296)
```

Words on Platform 2

Stepping through text with arrow keys.

When you step downwards and out of the window the first time, the slider doesn't follow the cursor. If you find the cursor again, using the slider, and then step upwards then the slider will follow the cursor.

When you edit the text resulting in creating of a new page outside the window, the slider doesn't follow the cursor.

If, when the slider and cursor is "out of synch" steps past the end of the document, the following occur:

```
java.lang.ArrayIndexOutOfBoundsException: -1
  at AxLink.Z42(AxLink.java)
  at WPViewer.B00(WPViewer.java)
  at AxViewer.run(AxViewer.java)
  at java.lang.Thread.run(Thread.java:296)
```

It was easily repaired by exiting Words and restarting it from the Anyware Office applet.

Crashes on Platform 3

On loading HTML page(AAlogin.ax).

```
java.lang.NoSuchMethodError:Dlg.method<init>(LAxData;Z) not found
at Dlg.J01(Dlg.java)
at IASMain.B00(IASMain.java)
at IASMain.DeliverMessage(IASMain.java)
at IASMain.run(IASMain.java)
at java.lang.Thread.run(Thread.java:296)
```

On starting Words.

```
java.lang.NoSuchMethod C06 (Ljava/lang/string;LAxData;Z) not found
at AxViewer.X36(AxViewer.java)
at WPViewer.B00(WPViewer.java)
at AxViewer.run(AxViewer.java)
at java.lang.Thread.run(Thread.java:296)
```

Words on Platform 3

Pops up now and then.

```
java.lang.NullPointerException:bar
at sun.awt.motif.X11Graphics.<init>(X11Graphics.java:52)
at sun.awt.motif.MComponentPeer.getGraphics(MComponentPeer.java:156)
at sun.awt.motif.MComponentPeer.handleExpose(MComponentPeer.java:261)
at java.lang.Thread.run(Thread.java:296)
```

TRITA-IT R 97:21
ISSN 1103-534X
ISRN KTH/IT/R--97/21--SE



Telecommunication Systems Laboratory
Department of Teleinformatics
Royal Institute of Technology
Electrum 204
S-164 40 Kista
Sweden