Mifrenz: a new email client application for children

Dr Tim D. Hunt
Wintec
tim.hunt@wintec.ac.nz

Abstract

Products currently available for monitoring childrens' email usage are either considered to encourage dubious ethical behaviour or are time consuming for parents to administer. This paper describes the development of a new email client application for children called Mifrenz (http://mifrenz.com/). This new application gives parents the ability to let their children safely use email, with the minimum of intervention. It was developed using mostly free software and also with the desire to provide real first hand programming examples to demonstrate to students.

Keywords: Email, Java, Children, Parental supervision, SPAM, white list,

1 Introduction

For many adults, email has become the most common method of distance communication (and sometimes not so distant), despite the growing nuisance caused by unsolicited email – SPAM. Anti SPAM technology usually limits the quantity of SPAM to a level that still makes email worthwhile, at least for adults. This is not the case for children aged between 5 and 12 years old, where even one offensive email may be an unacceptable price to pay. If a parent wishes to let, or even encourage, their young children to use email, the safest approach is to provide continuous supervision of their email activities. Of course, finding time for this supervision may be an issue for many parents and so email use is either not supervised, or just does not happen.

A basic technique known as a 'white list' does, however, provides an excellent way for the elimination of nearly all SPAM (Templeton 2007). A white list is a list of email addresses from which email is accepted and all other email is automatically deleted. For many adults, a white list is not an acceptable solution to SPAM, as most adults expect and want to receive email from people who they have not yet added to their white list. For children, however, in theory, a white list provides an ideal method for parents to control who their children communicate with. However in practice, a parent must still find the time to create and maintain (add new friends) the white list in a timely manner – children will want it done NOW.

Another major issue exists with white lists; they can be turned off quite easily once the child starts to explore what all the buttons do. It is therefore evident that email client applications designed to be used by adults are not a satisfactory solution for parental supervision of email usage by young children.

Although a number of products are available to address some of these issues (see literature review) the author proposes new functionality that if present, with the best features of the available products would improve on the currently available solutions. This paper describes the design and development of such an email client, using Java (2007) and other free software development environments and tools. In addition, the work was performed in the context of teaching programming skills to undergraduate students and many of the design issues were tackled with teaching in mind.

2 Literature review

Very few studies have been conducted that solely focus on the use of email by children, however, the use of email by children has been included in numerous studies on general Internet use by children. These studies tend to focus on children of at least 7 years of age.

A recent survey by Livingstone and Bober (2004) of Internet use by children aged 9-19 in the United Kingdom found that 75% of them had access to the Internet from home: of those 72% used the Internet for sending and receiving emails. Among the regular users of the Internet, 4% "have sent a message to make someone feel uncomfortable or threatened", and 25% had received pornographic SPAM via email or instant messaging. 79% of the children said that they normally used the Internet unsupervised. Beran (Beran and Li 2005) reported that in the 12-14 age group, 21% of children said that they have been cyber-harassed multiple times and 3% said they had actually committed this type of harassment.

The benefits of Internet use by children aged 10-18 years from low income families has been studied by Jackson, von Eye, Biocca, Barbatsis, Zhao and Fitzgerald (2006). They investigated the academic benefits of home Internet use and concluded that the more the children used the Internet, the greater the positive benefit on their academic performance was. However, they found that children in that group made very little use of email and "chat tools". It was suggested that a major reason for this, was that children from low income families had friends of similar backgrounds and so were also unlikely to have Internet access – therefore no one to 'talk' to online.

This quality assured paper appeared at the 20th *Annual Conference of the National Advisory Committee on Computing Qualifications* (NACCQ 2007), Nelson, New Zealand. Samuel Mann and Noel Bridgeman (Eds). Reproduction for academic, not-for profit purposes permitted provided this text is included. www.naccq.ac.nz

It is interesting to see the conflict that exists between childrens' and parents' expectations of Internet access software. Whereas 69% of children do not want their parents monitoring their online activities, over 50% of parents want access to improved control and monitoring tools (Livingstone and Bober 2004). Monitoring of children seems to be a growing phenomenon (Delaney 2006) but the ethics of this are starting to be questioned in the popular media; for example an article (Hymowitz 2006) in the New York Times likened the practice of monitoring your child's email to reading their personal diary.

It is now possible for a parent to monitor a child's Internet access very closely (Delaney 2006):

> "Features vary among the different offerings, but most can record every keystroke a computer user types and log every Web site he visits. Many log incoming and outgoing email and instant messages and record "screen shots," or images of what the user does online. The software – including packages called Specter Pro, eBlaster, and IamBigBrother -- is designed to prevent children from tampering with its operation. It allows parents to see sites, including blogs, their children are accessing. The keystroke logging permits them to capture login and password information their kids have entered for blogs. That could allow parents to access even blogs that are password protected or are otherwise restricted."

2.1 Email clients for children

For young children, a different approach can be used to control email activity. Kidmail.net (Kidmail.net, 2007) offers a client application that is designed to appeal to young children (5-9 years) and also an email server service that controls (for an annual fee) the email a child receives. This is basically a 'white-list' solution to preventing email from unknown persons reaching the child. The parent (with help from the software) creates and maintains the white-list and also has the ability to read the child's emails. The software does not delete emails that are not on the white list, but gives the parent a chance to read them and add them as a new contact to the white list.

Starfish Family Mail (from Lincoln Beach Software Ltd, 2007) is an email client for children that uses a white list created by the administrator. The software can be setup so that any emails from addresses not on the white list, are either hidden from the child, or they require parent intervention to be read. The software also allows a basic authentication method to make sure that the email hasn't been spoofed (see next section). This product requires a high parental involvement to keep the white list up to date.

2.2 Spoofing

A potential flaw with relying on the white list to block SPAM is the concept of spoofing (Wikipedia 2007). An email is said to be spoofed when the 'From' email address has been deliberately changed so that the email appears to be from someone other than the true sender.

A very basic way to spoof an email is to use a false 'From' address. This type of spoofing (see Table 1) is easy to detect. The table shows some of the header fields of an email that was sent from a Hotmail account to a Gmail (2007) account. The sender was pretending to be a friend called Nice Person (nice.person@happyserver.com). Although easy to detect, GMail did not pick this up as SPAM, presumably because for adults at least, this is not a definite sign of spoofing.

from	Nice Person <nice.person@happyserver.com>
reply-to	Nice Person <nice.person@happyserver.com>
to	timhot@gmail.com
date	Mar 3, 2007 1:51 PM
subject	Let's meet at the park
mailed-by	hotmail.com

Table 1: This shows the detailed email header information from a spoofed email. Notice that the 'mailed-by' value is different to the 'from' and 'reply-to' values

Another method to reduce spoofing is to employ a 'hand shake' between the sender and receiver. Riyadh (Riyadh 2001) proposes the 'Correspondence Negotiation Protocol' to give more control to the receivers of email. The Anti-Phishing Working Group (2007) provides a comparison of various proposed methods of preventing email spoofing including checking the IP address of the sender and using digital signatures. The problem that they are trying to solve though is not the same as the problem for children, where email to children can be limited to an approved white list rather than trying to allow all non spoofed emails through.

2.3 Literature review conclusion

The use of email can provide a valuable learning experience for young children, yet also presents a risk to those same children. Many parents are turning to spying on their children's online activities and this poses ethical considerations and dilemmas. Two less intrusive solutions were identified based on the use of a white list; however neither solution provided all functionality in a low cost product.

3 Aims of this work

The currently available email clients do not provide a satisfactory and low cost solution for protecting young children from SPAM. This work aimed to address this gap in the market with a new email application specifically designed with both children and parental supervision in mind. As identified in the literature review a number of issues needed to be addressed.

1. The conflict between a child's need for privacy and the parent's responsibility to protect the child.
2. How to give a child the ability to email their friends in a timely manner while not requiring the parent to give constant supervision.
3. Keeping the cost of the solution low enough to allow widespread use.

In addition the following additional aims were identified:

4. Create a product that is both fun and simple to use for young children so they would want to use it.
5. Use techniques with the Java programming language that the author teaches in a number of computer programming courses so that they can be demonstrated to students.

The literature review identified that Internet use by young children is a potentially useful tool for improving their academic performance. By addressing the above aims, it was hoped that this work would create an application that does indeed encourage children to read and write (type) more than they otherwise would and therefore potentially improve those skills.

4 Design

4.1 Design overview

A new email client application called Mifrenz was created to meet each of the aims identified above. An overview of how each aim was addressed follows.

4.1.1 Conflict between a child and parents needs

When a parent logs on to the software, they are presented with the 'parent' window and they create the initial contact list which acts as a white list. The parent window does not display any emails that are sent or received. When the child logs on, they are taken to the 'child' window and can then email anyone on the contact list and receive email from anyone on the list. The parent is thus able to control the childrens contacts without being encouraged to read the email. Due to the young age of the children, the parent can easily view the child's password so they can retell it to the child without having to reset it. So, although the parent could logon as the child, the software does not encourage this behaviour and the parent would have to make a conscious decision to do so.

4.1.2 Timely manner

Once the initial list has been created, the child can then create new contacts himself or herself. The parent is automatically sent a 'contact approval request' to their normal (for example work) address, to which they only have to reply with a yes or no, to activate or block the new contact and any current or new emails. During the time interval between a child creating a new contact and the parent either approving or declining the contact, the contact is in a 'pending' state. While in this state, the child can use the new contact as normal, except when they press the send button, the email is held back (pending) behind the scenes, until the parent approves the contact. Emails received from a pending contact are also held back out of view until the contact is approved. However, if the contact is declined, all pending emails to and from this contact are deleted.

4.1.3 Low cost

Mifrenz uses a third party email service, initially Gmail to send and receive email, but all 'intelligence' is on the client application. This avoids the need to provide a server to supply the email service and so keeps the potential cost of the product low. Also, Mifrenz does not provide the extensive 'spying' capabilities of many other products and so the development time was relatively low.

As far as possible, Mifrenz was developed using free software development tools and environments. The main development environment was Netbeans (2007) and Java both free to use. To distribute the software, a website was created using the free Netbeans plug-in, Visual Web Application (Netbeans 2007) which allows WYSIWYG (What You See Is What You Get) web page development. Sun's (Sun Microsystems 2007) free "Sun Java System Application Server" provided the server platform and "Java Web Start" (Sun Microsystems 2007b), a relatively new and free technology, was used to distribute and install the software to users. MySQL (2007), a free database, is used to keep track of who downloads the software. Also various other free utilities are used to maintain the web server remotely, for example LogMeIn Free (LogMeIn.com) a remote control tool and FileZilla (2007) an FTP tool.

4.1.4 Fun and simple to use

Young children do not need the sophistication of many current email client applications (see for example Microsoft Outlook, 2007), and in fact the author believes that too much choice is an impediment to children using email. For example, young children only require an 'In box' and a 'Sent box', they do not need other folders to organise email. They do not need extensive information about each email such as date, importance, size etc.

Young children love jokes and collecting things. Mifrenz provides a simple way for them to create jokes and then to easily share them with their friends. The software automatically builds a joke list from jokes that are sent to them (or they can create new jokes), and they can then attach these new jokes to emails to share with others.

4.1.5 Teaching material

The author teaches Java programming and was keen to demonstrate actual use of the techniques being taught. For example: a) students are taught how to write objects to disk, and this is demonstrated by writing an email message object to disk, b) Threads are used to make the GUI interface more responsive by connecting with the email server on one thread, while a separate thread allows the child to write or read another email.

4.2 Major design decisions

The following sections describe some of the major design decisions made.

4.3 Three layer design

A well known design pattern is to separate the business logic of an application from the user interface and database/disk access code. This is typically implemented as three separate layers: the presentation layer, the business logic layer and the data access layer. Mifrenz used this pattern with each layer represented by a package of classes. Akin to a three tier client/server architecture design, each layer was designed to allow its possible replacement at a future date e.g. 1) all data flow and method calls from the presentation layer are to the business logic layer and not to the data access layer, and 2) the data access layer implements all calls to the disk storage and all connections to the email server.

An example of this design is seen when new emails are retrieved from the server. Although the dataAccessLogic package handles connections to the server, it is the EmailLogic class in the businessLogic package that determines if the email is from an approved contact and then determines what to do with the email.

Other examples of this design occur in the presentationLogic package which is responsible for all user interaction. Whenever possible, this package passes all decisions to the businessLogic package. This design solution decouples the GUI from the rest of the application.

4.4 Object oriented analysis and design

Object Oriented (OO) techniques are currently (Perks 2006) seen as 'best practice' for creating software and as Mifrenz is intended to be used as a teaching tool, OO techniques such as data encapsulation and inheritance were utilised throughout the process. For example, objects are used to store, manipulate and pass data and many of the interface classes inherit from one of the predefined Java classes.

4.5 The database design

The following considerations were taken into account when choosing a storage solution:

To keep the final cost of the product to a minimum, only free solutions were considered.

As Mifrenz is to be distributed via a website, a large database application such as MySQL (2007) would have resulted in an unacceptably long download time.

The author wishes to demonstrate techniques taught to his students which include object serialization, but not SQL databases.

Based on the above considerations, and because there are no concurrency and transaction management issues, it was decided to create a proprietary database using object serialisation techniques. In essence, each email is stored as a separate object on disk and is read into a List structure when the application is opened. The DatabaseWrapper class handles all read and writes to the disk, e.g. a method called getAllContacts(User user) returns a List of all the contacts for the user.

4.6 Email server

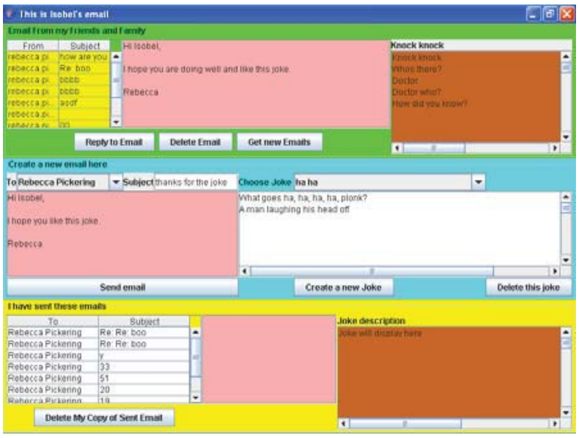
A decision was made to use a third party email server, so there would not be an ongoing service requirement. This keeps the cost of providing the product to a one off cost, rather than having to have a yearly fee as other similar solutions require. Initially Mifrenz was 'hard coded' to work with a GMail email account so as to reduce any setup issues and testing requirements. GMail was chosen as it is a global service, provides 2 GB of storage and allows POP email access. Before Mifrenz can be used, the parent must first create a GMail account for their child and enable it for POP access (instructions are provided to help do this). Mifrenz uses the Java email API to connect to the GMail email server allowing the sending and receiving of emails. GMail requires 'Secure Socket Layer' (SSL) access which the Java API also provides.

4.7 User interface design

Mifrenz has a separate user interface for the child and the parent. It also has an email notification for parents.

4.7.1 Child interface design

The overriding factor in this design was to make it as simple as possible. Again, the product was designed with teaching in mind, in that some basic design principles were followed e.g. the users focus naturally starts at the top left of the screen and moves towards the lower right. As mentioned earlier, it was felt that many of the features of 'normal' email applications were not necessary for young children, and therefore these were not included. Figure 1 shows the layout of the child interface. To keep it simple, there is only the one window (except for the 'pop up' joke creation window) that is split into 3 horizontal layers: The top layer is for emails that have been received, the middle layer is for creating new emails, and the bottom layer is for emails that have been sent.



From	Subject
rebecca pi	how are you
rebecca pi	Re: boo
rebecca pi	bbbb
rebecca pi	bbbb
rebecca.pi..	asdf
rebecca.pi..	

Hi Isobel,

I hope you are doing well and like this joke.

Rebecca

Knock knock

Knock knock
Whos there?
Doctor
Doctor who?
How did you know?

To Rebecca Pickering Subject thanks for the joke

Choose Joke ha ha

Hi Isobel,

I hope you like this joke.

Rebecca

What goes ha, ha, ha, ha, plonk?
A man laughing his head off

To	Subject
Rebecca Pickering	Re: Re: boo
Rebecca Pickering	Re: Re: boo
Rebecca Pickering	y
Rebecca Pickering	33
Rebecca Pickering	51
Rebecca Pickering	20
Rebecca Pickering	19

Joke description

Joke will display here

Figure 1: The child interface consists of a single window split into three horizontal layers for 1) received emails, 2) composing new emails and 3) displaying sent emails.

The top layer has 3 panels: the left panel shows a list of received emails, the centre panel displays the text of the email and the right panel displays the joke (if any) that was attached. Unread emails are differentiated from read emails by a different colour. There are 3 buttons for: 1) Replying to the selected email, 2) deleting an email and 3) for checking for new emails.

The middle layer is split into two panels: the left panel is used to compose a new email, and the right panel is used to choose and display a joke to attach to the email. The child chooses who to email by either pressing the 'Reply to Email' button, or by choosing the contact from the 'To' drop down list.

The bottom layer is basically a repeat of the top layer, but instead displays the emails that have been sent.

Scroll bars appear automatically when there is more information than can be entirely displayed in a panel. The GUI layout uses a Java technology known as 'Layout Managers' that provides windows that automatically change size to suit the size of the available monitor resolution and the information being displayed. This avoids the display problems that are inherent with fixed position by pixel design that many other applications use. This is another example of good design that can be demonstrated to students.

4.7.2 Parent interface design

The first time that Mifrenz is used, an Administrator Wizard runs automatically to assist the parent with entering the required information for initial setup.

To assist setting up, the overall instructions appear unchanged on each screen, while the specific instructions for each stage of the Wizard change as the user progresses through the Wizard. The instructions include details of how to configure a GMail account to allow POP email access.

Once the user has finished the Administrator Wizard, they are taken to the main Parent screen. This screen has 3 tabs that allow the parent to modify and add to any of the information entered through the wizard, namely 1) contacts, 2) children's details and 3) details of the email server that they are connecting to.

A button at the bottom of the Parent screen allows the parent to run the Administrator Wizard whenever they wish.

4.8 Parental supervision design

The author believes that it is not ethical for parents to read their children's email, but that it is a parent's duty to protect their children from SPAM email. The use of a white list is an ideal way to block email from all sources except from those on the list. The white list (i.e. the child's list of contacts) is imbedded into Mifrenz and there is no option to not use it. To prevent the child accessing GMail directly, i.e. using the GMail web site, the child's GMail password is hidden in the parent's password protected area of Mifrenz.

A child's friends are likely to be changing at a rate higher than a parent wishes to enter the new contact details, and certainly faster than a child is willing to wait until their parent gets around to entering the new contacts. Mifrenz was designed to make it as easy as possible for parents to vet their child's new contacts and update the white list. This is achieved by allowing the child to create the new contact themselves, and even write and 'send' the email before their parent approves the contact. However, 'send' is not what actually happens, as that email is held back until the contact is approved. A design was sought that would allow this approval process to occur as quickly and as easily for the parent as possible – parents should not be required to use the same PC as the child (as one reviewed product required). As the design of Mifrenz did not allow for a central server, this approval process could not be achieved by logging on to a server, as required by one of the other products reviewed. Instead, when a new contact is created by a child, an email is sent to the parents specified email address. The subject of the email is "Contact request from Mifrenz" so that the parent can easily prioritise these requests if they want to. The email contains the name and email address of the new contact and asks the parent to reply with either yes or no on the first line of the body of the email. This email is then returned to the child's GMail account and is downloaded to Mifrenz the next time the child checks their email. Mifrenz intercepts this email by looking for the string "Contact request from Mifrenz" in the subject line. Mifrenz then processes the email, determining if the contact has been approved or declined. If the contact has been approved, Mifrenz then checks to see if there are any emails waiting to be sent to this contact and either sends them, or deletes them if the contact request was declined.

This design allows a parent to help their children choose who they send and receive emails from. The parent has the opportunity to discuss with their children who their new friends are and how they met them. The child knows that their parent is being asked to approve their new contacts and no deceit is happening. Of course, a parent could still go and read their children's email by either logging on to Mifrenz with the child's password, or by using the Web interface of GMail. Just as the author believes a parent should not be reading a child's diary, it is up to the parent to refrain from snooping on their child's email activities. Most of the other email supervision products that have been identified seem to actively encourage a parent to snoop on their child's emails.

4.9 Robustness of design

Although Mifrenz is not a mission critical application, its design had to be robust enough to not frustrate either parents or their children; it certainly aimed to have a 100% guarantee of not letting through emails that were from addresses not on the white list. It is worth pointing out, that Mifrenz also benefits from the anti SPAM tools deployed by GMail, although this gives no guarantee of completely eliminating SPAM. In addition to keeping the users happy (by not behaving unexpectedly), Mifrenz was also created as a teaching aide and so was required to demonstrate good design practice.

Java in itself is an excellent language for robust design and so a good foundation to build on. Extensive use of the try-catch-throw structure was used with errors thrown to the business logic layer when ever possible, thus keeping decisions in this layer.

4.9.1 Spoofing

The reliance on a white list to block unwanted emails still leaves the potential for receiving spoofed emails. See the section on spoofing in 'Future Work' later in this paper.

5 Future work

A rather long list of fixes and functionality additions already exits. For example the GUI's are currently very basic and would benefit from tooltips and icons on each of the buttons.

6 Summary and conclusions

This paper has described the development of an email client application for children – Mifrenz. The application was developed to provide a low cost, low maintenance and ethically acceptable way for parents to guide young children's use of email and so protect them from the risk of receiving harmful SPAM.

The application is currently in a working beta stage and is available for free download from http://mifrenz.com/

7 Acknowledgements

Rebecca and Isobel – great testers – thank you.

The author would like to thank Wintec for permission to publish this work and for funding to attend the conference.

8 References

The Anti-Phishing Working Group (2007), Proposed Solutions to Address the Threat of Email Spoofing Scams, n.d., viewed 3 March 2007, http://www.antiphishing.org/Proposed%20Solutions%20to%20Address%20the%20Threat%20of%20Email%20Spoofing%20Scams%20White%20Paper.pdf

Beran T. and Li Q. (2005): Cyber-harassment: A study of a new method for an old behavior. Journal of Educational Computing Research, Issue: Volume 32, Number 3 / 2005 Pages: 265 - 277

Delaney K. (2006): Spy vs. Spy: Tools for Shadowing Teens Online. Wall Street Journal. (Eastern edition). New York, N.Y.: Nov 26, 2005. pg. A.6

Extreme Programming, n.d., viewed 28 February 2007, http://www.extremeprogramming.org/

FileZilla, n.d., viewed 28 February 2007, http://filezilla.sourceforge.net/

Geocities, n.d., viewed 28 February 2007, http://geocities.yahoo.com/

GMail, n.d., viewed 28 February 2007, http://gmail.com

Hymowitz K.S. (2006): Taste: Big Mother Is Watching. Wall Street Journal. (Eastern edition). New York, N.Y.: Aug 4, 2006. pg. W.9

Jackson L.A., von Eye A., Biocca F.A., Barbatsis G., Zhao Y. and Fitzgerald H.E. (2006): Does Home Internet Use Influence the Academic Performance of Low-Income Children? Developmental Psychology 2006, Vol. 42, No. 3, 000–000

Java.net, CollabNet, n.d., viewed 28 February 2007, https://java3d.dev.java.net/

Java SDK 1.5 The Sun Developer Network, n.d., viewed 28 February 2007, http://java.sun.com/

JUnit n.d., viewed 28 February 2007, http://junit.sourceforge.net/

Kidmail.net, n.d., viewed 28 February 2007, http://www.kidmail.net/default.asp

Lincoln Beach Software Ltd, n.d., viewed 4 March 2007, http://www.lincolnbeach.com/sfm_main.asp

Livingstone S. and Bober M. (2004) UK children go online : surveying the experiences of young people and their parents [online]. London: LSE Research Online, viewed 28 February 2007, http://eprints.lse.ac.uk/archive/00000395

LogMeIn.com, LogMeIn Free, n.d., viewed 28 February 2007, https://secure.logmein.com/home.asp

Microsoft Outlook, Microsoft Corporation, n.d., viewed 28 February 2007, http://office.microsoft.com/en-au/outlook/default.aspx

MySQL Server 5.0 MySQL AB Co. n.d., viewed 28 February 2007, http://www.mysql.com/

Netbeans IDE 5.5, n.d., viewed 28 February 2007, http://www.netbeans.org/

Perks M. (2006): Best practices for software development projects, IBM Software Services for WebSphere, 10 August 2006, viewed 6 March 2007, http://www-128.ibm.com/developerworks/websphere/library/techarticles/0306_perks/perks2.html

Riyadh M.M. (2001): Towards security and balance in email through correspondence negotiation. Master of Science Thesis, School of Computer Science, University of Windsor, 2005, viewed 28 February 2007, http://ir.lib.sfu.ca:8080/retrieve/2180/etd1827.pdf

Sun Microsystems, Sun Java System Application Server, n.d., viewed 28 February 2007, http://www.sun.com/software/products/appsrvr/index.xml

Sun Microsystems, Java Web Start, n.d., viewed 28 February 2007b, http://www.sun.com/software/products/appsrvr/index.xml

Templeton, B., A variety of weapons against SPAM that don't require laws, n.d., viewed 4 March 2007, http://www.templetons.com/brad/spam/spamfix.html

Treehouse NetWorks Ltd, n.d., viewed 28 February 2007b, http://treenetnz.com/

Wikipedia, E-mail spoofing, 10 February 2007, viewed 1 March 2007, http://en.wikipedia.org/wiki/E-mail_spoofing