An architecture for WebHelp analysis and development: A case study

Peter Busch
Macquarie University

Francois Latouche
Macquarie University

Follow this and additional works at: http://aisel.aisnet.org/acis2005

Recommended Citation
http://aisel.aisnet.org/acis2005/10

This material is brought to you by the Australasian (ACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ACIS 2005 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.
An architecture for WebHelp analysis and development: A case study

Francois Latouche and Peter Busch

Department of Computing
Macquarie University
North Ryde 2109
Email: busch@ics.mq.edu.au

Abstract
This paper reviews the issues faced by developers using current technologies to implement web applications and examines these issues in the context of more modern tools such as Web Services and Rich Internet Applications. The paper explores the case of a particular company, briefly discussing its history, evolution and some of the requirements that have made a web based solution superior to available alternatives, ultimately leading to the development of a three tier helpdesk web application. The application’s development, functionalities and longevity are then reviewed and critically analysed with the aim of identifying flaws with its underlying architecture and the technologies employed to develop the system and other similar web based solutions.

Keywords
Webhelp system; Practical issues; Internet; WWW; Commercial applications; Security; Integrity; HelpDesk; Three tier architecture

1. INTRODUCTION

The rapid proliferation of the Internet and more specifically the World Wide Web has revolutionised the way people within most developed societies live their lives today (Chen 2001). Everything from the most mundane tasks such as checking the weather or public transport timetables to more complex and serious aspects of our lives such as banking and buying property can be done at the click of a button (Tanenbaum 2003). If the statistics are correct over 600 million people from around the world were regularly using the Internet towards the end of 2002 (Anon “How many Online?”). This figure is estimated to grow to 900 million by the end of this year (Anon “Internet User Forecast by Country”), which makes for a 50% increase over a period of 2 years. This revolution has not gone unnoticed by business organisations and the past few years have seen a great increase in the number of services and applications developed specifically to function over the Internet. The ‘anywhere, anytime’ concept of the Internet makes it the perfect medium to facilitate business and communication between organisations and people across the world making issues like location and time zones a problem of the past (Chen 2001).

2. COMMERCIALISATION OF WEB APPLICATIONS

The last ten years has seen a rapid evolution of web based application spanning many technologies, from simple, static web pages to fully interactive sites. However, it quickly became obvious to companies offering online services that no matter how advanced their web applications, they still could not offer a fully automated service to their customers without the cooperation of their business partners, suppliers and other concerned third parties, hence giving birth to the notion of e-commerce. “For e-commerce, companies need to display product listings and availability, take orders, accept credit cards, schedule deliveries, track user contact information, and so on. This requires much more that static web pages; it requires Web applications…” (Kurata 2002 :1). The concept of e-commerce introduces the idea of an integrated infrastructure spanning many companies and application systems, for example, a customer ordering a book online would require the participation of the book store itself but also a bank to accept and process payment details and a courier company to deliver the book. While the book store offering the online service provides the user interface required for placing the order, its underlying system also needs to be able to interact with the bank’s and courier’s systems seamlessly in a manner that would be transparent to the user. The integration of different systems running on different platforms has always been an extremely hard task to tackle until the advent of web services which purports to resolve the most common integration issues in a simple and efficient manner (Tabor 2002).

3. EVOLUTION OF WEB TECHNOLOGIES -THREE TIER ARCHITECTURE

As the novelty of static web pages started to wear off, users demanded better web sites with up-to-date information and new technology was required to address the current limitations resulting in the development of a number of technologies such as CGI scripting, CSS, client side scripting, applets and servlets among others. Technologies such as CGI scripting introduced the notion of interactivity in the context of web applications but also open new doors to hackers. “The scripts are seductively easy to write - but not so easy to write well. One small mistake in a script exposes the web server and its host machine to attack by malicious intruders” (Stein 1998 :313). The answer to these shortcomings came in the form of three-tier web applications, a concept that was borrowed from operating system centric applications to resolve issues identified with older monolithic and two-tier applications (Willet 1997). The three-tier architecture (figure 1) is extremely popular today and is implemented in many flavours in the web development arena; the two most common ones being Microsoft’s Application Server Pages (ASP) and Sun’s Java 2 Enterprise Edition (J2EE) which encompasses a number of technologies including Java.
Other well known tools competing for market share in this space are PHP and Coldfusion. Whilst the development tools might be different, the concept behind those technologies is essentially the same. The three tier architecture focuses on the clear demarcation between the user interface, the business logic and the back-end databases (Anderson 1998).

4. FLAWS OF CURRENT TECHNOLOGIES

There is no doubt that while web technologies have undergone tremendous growth and technological advances over the past two decades, there are still a number of areas that need to be addressed before these technologies can be considered fit for developing fully fledged and totally autonomous commercial applications. The following issues have been deemed worthy of mention here because most of them are well proven in the client server arena and will have to be resolved if web applications are to compete with the older, more stable technologies.

4.1 The web application user interface

Whilst web applications' user interfaces have come a long way since their inception, through the use of technologies such as CSS, client side scripting and applets, there is still a lot of room for improvement in this area (Nielsen 2002). All of these technologies seem to focus on one aspect of the user interface design, with CSS focusing of appearance, client side scripting on functionality and applets on providing a basic development environment. This lack of integration among current technologies and their unreliability when it comes to version control results in user interfaces still being an area of unresolved issues (Nielsen 2002).

4.2 Protection of intellectual property

Apart from later technologies such as applets and servlets, most business logic residing on an application server is currently held as non-compiled scripts (Kurata 2002). Such an approach is acceptable if the application is to be used internally as most of these scripts are not sent to the client and are therefore hidden from users. However, if an application is developed with the intent of being marketed and sold to third parties there is currently no way of protecting intellectual property. Some of the technologies such as Coldfusion provide a way of encrypting the code contained within the pages held on the application server, but a quick search on the internet will return different programs capable of cracking the encryption. The best solution here is to be able to deploy a compiled version of the application on customer sites and keep the source code for internal use. While Java technologies such as applets and Servlets provide this capability, the next point rules them out.

4.3 Integration

The advent of e-commerce and the notion of ‘supply chain’ have demanded a level of system integration that none of the technologies discussed so far have been able to provide (Tabor 2002). While the solutions discussed above could all be used to design perfectly robust stand alone applications, they do not provide any easy way to transfer data from an application developed using a specific technology to another (based on a different technology), even applications derived from the same development tools necessitate a substantial amount of time being vested into the careful design of rules and standards in order to facilitate the transfer of data.

4.4 Lack of Security

The lack of security features offered by many of the earlier technologies has also meant that many companies implementing web applications have had to rely of third party software to secure their applications (Stein 1998). While this practice is acceptable, it is clearly not ideal and greatly reduces the overall manageability and autonomy of the application. The latest
security features made available by many of the technologies discussed above are now mature and dependable; however, if one is to redesign an old application or implement a new one, then all the pros and cons outlined above would have to be taken into consideration in order to identify the best possible solution.

4.5 State Management

Unlike desktop applications which operate within the confines of well-established local area networks, web applications operate on a global scale and as such have to deal with a multitude of technical issues which do not apply to most client/server applications. One of the biggest issues facing web application developers today is that of state management. While HTTP, which, as mentioned before, is the underlying transport protocol for the web and is extremely well adapted to traditional browsing, it is essentially stateless (Bradbury 2004) resulting in great difficulties when it comes to keeping track of users. However, keeping track of users’ interaction with the system is at the heart of any application and is crucial to any transactional system. A number of approaches have been used to introduce a notion of state management to web applications with one of the most commonly advocated today being the use of cookies (Stein 1998), which work by storing basic information on the user’s machine. When the user revisits a site this information is read and sent back to the server which uses it to identify the user. While this approach and others like URL parameters, have been adequate they are not fully reliable. The use of cookies is somewhat limited when one considers that each individual user is free to suppress them on their machine. While most web applications have, in one way or another, overcome the issue of state management, most of the approaches used today still rely on some criteria to be met on the client side to work properly, a notion that is adverse to the thin client architecture.

5. COMPANY X

Company X was first created in 1997 with the primary aim of providing consulting services focusing on the implementation of Enterprise Resource Planning (ERP) information systems. Having occupied high-level positions within a manufacturing company that was using one of the big ERP packages of the time, the owners and employees of Company X all had extensive experience in a number of areas of that particular package. Their combined expertise spanned over, but was not limited to, areas such as finance, sales and purchasing, services, manufacturing and technical aspects of the system. The decision to form Company X was primarily based on an apparent gap within the level of support provided by the then vendor of the ERP package resulting in a high level of dissatisfaction from their customer base. The latter complained that the software vendor was just too big, did not understand their business and could therefore not provide them with the level of support they required to manage such a complicated system. The owners of Company X identified this as a niche requirement and decided to work closely with the software vendor and target customers’ issues. Company X quickly gained a reputation among the customer base as a high quality support provider by forging a strong partnership with the software vendor and building a rapport based on trust with its customers essentially becoming the surrogate implementation arm of the software vendor.

5.1 Company’s Evolution

Company X was now responsible for assisting customers with implementing the system. This involved identifying and rectifying, where appropriate, their current business processes, using modeling tools and techniques to ensure the best system fit within the organisation and carrying out system customisation to accommodate special business requirements. Once the implementation was complete, Company X took over the day-to-day customer support and in some cases even provided long-term experienced technical staff to customers to manage and administer their system. As demand for Company’s X services grew to overseas customers, more precisely those in New Zealand and South East Asia, there was a need to expand to effectively address these needs. While Company X has had to recruit new staff over the years to address increasing demands, its owners have resisted the temptation to grow too quickly and have so far remained focused on the Company’s core objective which has made it successful in the first place and small enough to provide a ‘personalised’ level of service and support to its customers. Today Company X has fewer than 10 employees and is fully owned by its two directors.

5.2 Reassessment of Business Strategy

By early 2000, two factors had come into play, which started a period of change and, to a certain extent, uncertainty for Company X. First, the technology behind the ERP system, which had so far been its main source of revenue, was starting to age with customers becoming aware of better systems and beginning to turn elsewhere for more technologically advanced solutions. The second aspect that heavily impacted on business was that around the same time, the manufacturing industry in Australia was starting to slow down with many companies looking at relocating their operations to more cost effective countries in Asia, thereby greatly reducing the demand for ERP systems.

Faced with the sudden turn of events, the owners had to reassess their position in the marketplace and four main plans of action were put in place: First, customers who had invested a substantial amount of money in the old ERP system were unlikely to look elsewhere in the near future and would still generate a lot of work in the form of day-to-day support, new projects and system customisation. Enough resources would have to be allocated to ensure that these customers continued receiving the level of service they had become accustomed to. Second, strategic partnerships had become a buzz word and it made sense in light of the broad array of combined expertise within the IT industry of Company’s X staff and the number of solid relationships that had been forged over the past few years, to look into new product lines. The decision makers decided to add a reseller arm to the Company. A few software packages that fit within Company’s X core expertise were identified and added to the company’s portfolio. Third from an application point of view, most of the Company’s employees had so far focused on ERP type applications as this had historically been the Company’s core business. It was decided that this should
be expanded; analysts, application consultants and implementers were retrained and given exposure to newer systems like Customer Relationship Management (CRM) and Supply Chain Management (SCM). Fourth, the Company’s technical staff and developers had spent most of the last few years working on antiquated technology and it was decided that they also should get retrained over the next few years to come up to speed with newer technological advances, more precisely web technologies which had become extremely popular.

6. IMPLEMENTATION OF A HELPDESK SYSTEM

Company X had been approached by one of its clients to jointly develop a helpdesk system, the primary purpose of which, from the client’s point of view was to provide its employees with an easy, efficient way of logging issues encountered while using the ERP system on a day-to-day basis. The client was a multi national with branches around the world. Branches in the Asia Pacific had recently implemented the ERP system outlined above. Because of the great number of users, the segregation of the user base and the language barrier, it had been extremely problematical to conduct a complete and thorough training schedule for all users and productivity was suffering as a result of this. Company X could also see the benefits of implementing such a system, by putting it at the disposal of its customer base and employees; the company would effectively improve its own efficiency. Customer management had often been undertaken in a haphazard manner with each individual consultant being responsible for following through after initial discussions with customers. This made managing resources somewhat difficult. Essentially, the system would be used as a central repository of information that could be used to ensure a timely response to customers’ requests, as well as providing a documented history of user issues and resolution, a knowledge base and a way to manage internal resources.

6.1 System’s Background

One of the main considerations behind the design of system was an attempt to resolve the issue of having to deal with a user base that spanned many countries and time zones. This made it extremely frustrating for users who often could not get their issues resolved in a timely manner. The main support centre was based in Australia but due to time differences a substantial proportion of the user base did not have access to the relevant support resources when they needed it. Furthermore many of the users were from countries where English was the second language. The other major issue involved there being no central repository of information. First, the fact that there was no structured way of storing users’ issues and their resolutions often resulted in unnecessary duplication of work. Second, the nature of a call also often resulted in the call having to be reassigned to a more appropriate support person than the one who originally took the call. Finally, because of the informal support structure and lack of cohesion, users found it hard to keep track of logged issues often not knowing who was currently working on them and when they could expect a resolution. It was therefore decided that the system would have to provide at a minimum: (1) Easy accessibility: The intention was that any PC with a connection to the Internet should be able to access the system, allowing users to log calls as issues arose. (2) A simple user interface: As one of the aims behind the system was to reduce user frustration, it was very important for the system to provide a simple user interface that would allow users to efficiently log their calls. (3) Accountability and traceability. By giving user access to the issue’s history, the latter would be able to keep track of exactly who was responsible for their call and the current status of the call. (4) The Knowledge Base: It was vital that the system should provide appropriate search facilities to both users and support people to research the issue at hand and quickly identify potential solutions. And finally (5) call management: The system would effectively take over the more simple call management tasks essentially freeing support people to focus on the issue at hand. This would primarily involve, but would not be limited to, automatically notifying all the appropriate people when a call is logged, updated or re-assigned.

7. TECHNOLOGY AND SYSTEM INFRASTRUCTURE

One of the systems criteria identified was the need for a system that would be easily accessible from any location. While initial investigation included the possibility of developing a normal client/server system accessible through VPN, it was clear that the most feasible solution, both from a financial and technological point of view, was a web-based application. With this in mind, an appropriate web development technology had to be selected for the job and because some of the technical people who would be involved with the project had some prior exposure to Coldfusion, the decision was made to use it for the implementation of the system. The second, and somewhat simpler, technological consideration was choosing the database. Because SQL server was already being used in conjunction with other application, it was the obvious choice. A three tier architecture ended up being adopted (figure 2), which included the following layers. Firstly a Client or Presentation Tier which was implemented using HTML and Javascript running on the client’s browser.
Secondly a Business Tier which in turn consisted of four main components: (1) The Internet Security and Acceleration Server: While this component is not directly involved in the implementation of the system, it is still an integral part of the overall architecture and, as such, should be mentioned here. All traffic going in or out the domain passes through the ISA server. (2) The Authentix Server: Authentix is third party software used to ensure that only authorised users can access the system. While Authentix can be configured to use cookies, it is currently set to secure access by protecting the directories within which the HTML pages reside. Any user trying to access pages within the protected directories through HTTP will be prompted for a username and password before being allowed to proceed any further. (3) The IIS Server: The IIS server is the first component where actual business logic processing takes place. The IIS server is responsible for retrieving HTML pages requested by the client and either forwarding the request to the Coldfusion server for processing or back to the client. (4) The Coldfusion Server: The IIS server will forward and HTML page containing embedded Coldfusion code to the Coldfusion server for processing. The Coldfusion server will in turn interpret the code and act accordingly, where necessary interrogating the database. Once processing is complete, the Coldfusion server forwards the result to the IIS server, which in turns forwards the resulting HTML page to the client. While the business tier involves a number of components at a logical level, all these are implemented on two physical servers. First, the Firewall server runs ISA and is essentially the gateway to the internal domain and demilitarised zone (i.e. DMZ). Second, the web server, which sits in the DMZ, runs the IIS, Authentix and Coldfusion servers. Finally the Data Tier with SQL server provided the persistent data container for the application and sits within the domain.

8. SYSTEM IMPLEMENTATION

8.1 Functionality
The system has so far been kept simple, easy to use and true to its core purpose. The system provides two levels of access, one for normal users and the other for support/administration users.

8.2 Normal Users
A normal user is one that has limited access to the system, their access is restricted in two main ways; first, they have no administrative power and second, they can only view calls that are relevant to them. The main functionalities available to normal users are as follows: Firstly, logging a new issue: This is where a new call is logged in the system. This system will default their user information such as name, email address and phone number and allow them to provide information about their issue. Secondly, viewing Issues: To ensure customer privacy, normal users can only view issues that relate to their own company. Users can view all of their company’s call currently awaiting resolution and the ones that have already been closed. Thirdly, updating issues: In addition to viewing open issues, normal users can also update calls that belong to their company. In the same way as when a call is first logged, an email will be triggered to the appropriate people whenever an existing call is updated. Fourth, searching the system: There are four ways of searching the system: (1) By call number – using the field available on the left hand side of the screen. (2) Searching the ‘calls’ table by keyword – Figure 3 illustrates the simple, yet functional search screen which goes through the main ‘calls’ table and searches the selected fields for the specified words. (3) Searching the ‘calls’ table by characteristics – Figure 4 shows the search screen, which goes through the main ‘calls’ table and identifies all records that match the specified criteria. Here all calls of the same status or assigned to the same person can be identified. (4) Searching the Knowledge Base: The system also contains a knowledge base, which can be populated either
manually or by converting an existing call. The knowledge base can be searched by keywords much the same way as the ‘calls’ table. (5) Closing and reopening calls: A user can close or reopen any of their company’s call, an action, which once again, will trigger emails to be sent to the appropriate people.

![Search Interface](image)

Figure: 3 Searching for calls by Keyword

8.3 Support/Administrative Users

All the functionalities listed above are available to support users, but the latter also have access to the system’s administration area and, unlike normal users who can only view calls relating to their companies, support users can view and update all calls currently in the system. The system administration area allows users to: Modify the content of a call: This is useful when a minor fix or modification needs to be done to a call which does not require emails to everyone related to the call. Maintain Products: Products against which calls are to be logged can be defined here. Maintain Status: This is where valid call status’ are maintained. Maintain Initial Support: As mentioned above, whenever a new call is logged, an email is sent to a support person who is then responsible for assigning the call. This area of the system is used to define who these initial people are. Support people can be defined here based on a combination of company and product. For example, one person might be responsible for any call logged by users belonging to company A that relates to product A, but another person could be made responsible for all calls by the same company that relate to product B. Maintain Support Staff: This is where support people are setup. Information such as the person’s username, name and contact details are maintained here. In order to ensure a correlation with the Authentix username, which is used for authentication, the username defined here has to match the name defined in Authentix.

Maintain Clients: This is where normal users are setup. Information such as the person’s username, name and contact details are maintained here. In order to ensure a correlation with the Authentix username, which is used for authentication, the username defined here has to match the name defined in Authentix. Maintain Enhancement History: The wide segregation of the user base means that it is hard to inform all system users when new functionalities are implemented, therefore modifications and enhancements to the system are recorded here. This documentation then becomes available to users and can be used to determine how to use new system features. Transaction Logs: The system currently keeps track of every connection to the server and support people can review these logs. Maintaining the Knowledge Base: This is where information is manually entered into the knowledge base.

9. CURRENT SYSTEM FLAWS

9.1 Current User Interface Flaws

The current system interface is fairly basic, relying primarily on HTML for presentation. All the main components including text boxes, drop down boxes, check boxes etc. are currently implemented through HTML. Special effects, such as inverting colours when the mouse is moved on a menu item for example, are achieved by using a combination of Cascading Style Sheet and Java scripting. Javascript is also used to carry out user input validation on maintenance screens such as the ones used to log or update calls and administer the system. There are a number of user interface flaws, which begin with the system lacking fluidity: While HTML is simple to use and allows for rapid development of user interface it is very rigid in structure and lacks the full array of features that makes client/server applications so user friendly and intuitive. Most client/server application allows users to quickly access multiple source of information often in ‘an easy to use’ format, being based on HTML the current helpdesk system only ever makes one set of data available to the user at a time making it very cumbersome and somewhat inefficient especially when compared with many 32-bit windows applications. Also Java scripting can be bypassed: Most browsers allow users to choose whether to permit JavaScripts from being executed, essentially resulting in the potential suppression of client side validation, thereby rendering them useless. This means that in many cases the system’s data integrity could only be ensured by enforcing the same validations on both the client and server resulting in unnecessary duplication of processing and reducing the system’s performance. Finally with regard to the latency: Because the current system uses HTML, it is limited to the confines of the technology and as such, can only ever store and display one set of data on the client side meaning that any new data requested by users need to be retrieved from the server. This makes for a very time consuming process especially if the connection to the Internet is already slow. Client/server applications usually work much more efficiently by making better use of the client side resources.
9.2 Inability to protect intellectual property

The system was developed using Coldfusion in the middle tier to implement the business logic and while this choice was valid at the time due its relative simplicity, this choice is now proving to be a hampering factor when in comes to protection of intellectual property. Coldfusion, just like JSP and ASP, is a scripting language and as such it is embedded within underlying HTML pages comprising the system. This essentially means that installing the system on a third party’s site would result in them having full access to the source code upon which the system is based. While Macromedia has supplied an encryption utility to address this problem, it was obvious that it was not quite up to the task.

9.3 Current Security Flaws

There are two main ways of accessing the system, first through the main URL which takes the user to the system’s main page and second by clicking on a link contained within a system generated email triggered by an update to a call, this link takes the user directly to the call described within the email. However, credentials must be provided prior to gaining access to the system, as mentioned above, this is implemented through Authenticix. System security is not currently implemented from within the application itself but by monitoring HTTP access to the directories within which the system’s HTML pages reside. Security setup is done through the Authenticix security manager. Once a directory has been defined as protected, user names and password have to be set up to allow HTTP access to these directories; this can be done selecting the appropriate directory and adding users to it. While, the above method of ensuring secure access works quite well, it also proves to be flawed in a number of ways: Convoluted user setup: The current procedure requires for a user to be setup in both Authenticix and the application. This obviously involves an unnecessary level of complexity and duplication of information. Limited session management: Because authentication is done outside of the application, session management is very limited. This results in the inability to properly monitor the number of connections to the system and makes licensing a very hard issue to tackle. Authentication process not secure: Currently the user name and password is sent to the server in clear text potentially exposing them to a ‘man in the middle’ attack. User setup cannot be done by HTTP: Because the user setup is currently done via the Authenticix manager, which is not available through HTTP, all user management tasks have to be done internally, a process that can be very time consuming. Had user management been more flexible, customers could have been made responsible for setting up and managing their own users.

9.4 Current User Interface Flaws

The current system interface is fairly basis, relying primarily on HTML for presentation. All the main components including text boxes, drop down boxes, check boxes etc. are currently implemented through HTML. Special effects, such as inverting colours when the mouse is moved on a menu item for example, are achieved by using a combination of Cascading Style Sheet and Java scripting. Javascript is also used to carry out user input validation on maintenance screens such as the ones used to log or update calls and administer the system. There are a number of user interface flaws, which begin with the system lacking fluidity: While HTML is simple to use and allows for rapid development of user interface it is very rigid in structure and lacks the full array of features that makes client/server applications so user friendly and intuitive. Most client/server application allows users to quickly access multiple source of information often in ‘an easy to use’ format, being based on HTML the current helpdesk system only ever makes one set of data available to the user at a time making it very cumbersome and somewhat inefficient especially when compared with many 32-bit windows applications. Also Java scripting can be bypassed: Most browsers allow users to choose whether to permit JavaScripts from being executed, essentially resulting in the potential suppression of client side validation, thereby rendering them useless. This means that in many cases the system’s data integrity could only be ensured by enforcing the same validations on both the client and server resulting in unnecessary duplication of processing and reducing the system’s performance. Finally with regard to the latency: Because the current system uses HTML, it is limited to the confines of the technology and as such, can only ever store and display one set of data on the client side meaning that any new data requested by users need to be retrieved from the server. This makes for a very time consuming process especially if the connection to the Internet is already slow. Client/server applications usually work much more efficiently by making better use of the client side resources.

9.5 Current Business Logic Implementation Flaws

Coldfusion provides developers with all the constructs usually associated with most of the well-known programming languages. It supports conditional statements, iterations and even provides programmers with a number of useful functions like ‘ListContainsNoCase’, which greatly reduce development time. But for all the benefits it brings to the table Coldfusion as a web technology falls short in a number of areas. For example with regard to Open source: The code listed above is simply embedded in the ‘head’ section of one of the system’s HTML pages and gets interpreted by the Coldfusion server in real time. This is not an issue when the system is deployed internally, since the Coldfusion code is never sent to the client. However it obviously represents a major problem in cases where the system needs to be deployed externally as the code would become available for everyone to see and use. Coldfusion provides a utility which can be used for encrypting an HTML page, however this solution is far from robust and would further compound the issue of latency.

Secondly it is an expensive solution: The helpdesk system described here is obviously not new in the market, but what sets it apart is its simplicity and ease of use, making it suitable for small to medium sized companies looking for a cheap solution that can be implemented quickly and requiring minimum ongoing management and maintenance. Unfortunately, the current requirement for a Coldfusion server, goes against this idea as it is relatively expensive to purchase and requires specific skills to install and manage. Finally the development environment is not very cohesive: The application is currently based on a number of technologies each responsible for a particular aspect of the system. Again this is not such a problem, if the system is only implemented internally in a controlled environment, however, it is definitely an important factor when considering
deployment in a new environment and introduces more variables in the mix when having to assess compatibility requirements.

9.6 State Management Flaws

Due to time constraints at the time of system development, a decision was made to develop two sets of user interfaces; one for normal users and the other for support people. While this greatly reduced development time by ensuring that programmers did not have to implement session based authentication, it is now proving to be a major issue from a system maintainability point of view. In many situations, this results in duplication of work with programmers having to apply the same fix/modification and testing to both areas of the system. By implementing state management from within the application, authentication could be managed in a much more orthodox manner therefore resulting in a more cohesive and self contained system.

10. POTENTIAL SOLUTION

After assessing the latest technologies available for the development of web applications, it was decided that the following technologies, whilst still weak in some areas, could resolve most of the issues described above.

10.1 Rich Internet Application

RIAs are similar to applets in that they are self contained client side applications that interact with users in real time by limiting the number of calls to the server. When a user connects to an RIA site, the application is first downloaded and executed within the context of the browser. Just like applets, RIAs often necessitate a plug in to be installed before they can be used, however, unlike applet developers; RIA developers have access to a rich integrated development environment which greatly improves efficiency and the quality of the user interface. In fact Steven Webster is quoted as saying that “RIAs boost end-user productivity by delivering a familiar and more productive user experience, which has an immediate impact on a business’ bottom line” (Bradbury 2004 :18). Because RIAs consist mostly of one file that is downloaded to the client for execution (Bradbury 2004), developers have to ensure that the size of the applications remain reasonable to ensure that it can quickly be downloaded, which in turn limits the usability of the technology. Another aspect that has yet to be proven as far as RIAs are concerned is that of compatibility, as was the case with applets, and because RIAs are in many ways based on similar principles, vendors will have to take appropriate measures to ensure that the technology does not go down the same road. In fact most suppliers have gone out of their way to emphasise the fact that their product works consistently across a number of browsers, operating systems and platforms, however, because the technology is relatively new, this remains to be confirmed.

10.2 Web Services

Web services have become the latest web application craze but unfortunately security still remains one of the greatest challenges. The attribute that makes web services so appealing i.e. accessibility, is also the main source of debate and reluctance when it comes to adopting the technology, in fact “71% of 500 respondent IT managers would not consider XML WS until security issues are resolved” (Anon. “InfoWorld Survey”). Securing web services is proving to be a major challenge and will undoubtedly determine the future success of the technology. Many major players including Microsoft, IBM and SUN are currently working on potential solutions, but unfortunately, unlike other aspects of web services which were developed jointly and abide to stringent standards, most of the security solutions submitted so far have been created independently resulting in a lack of cohesion (Chatterjee and Webber 2004), and nervousness on behalf of developers. While web services take care of implementing business logic on the server side, Rich Internet Applications (RIAs) are a relatively recent innovation intended to provide advanced user interfaces on the client side.

11. CONCLUSION

From a functional point of view the original helpdesk system was designed and developed to assist the company within which it was implemented to better manage their resources and improve their level of service. From a technological point of view, it provided a way to expose the company’s technical staff to emerging web technologies. Because the potential user base for the system was wide and varied it was very important to design a solution that would be easily accessible by most of today’s client machines. With this in mind, the system was implemented as a typical three-tier architecture application, which was very popular and well documented at the time of implementation. The system’s interface was implemented with HTML accessible through most of today’s browsers. A Coldfusion server was used to implement business logic and SQL server was used as the main system’s data repository.

The system was a great success from a functional point of view easily meeting, and in some instances, surpassing all of the main implementation requirements, these being; simplicity, accessibility, centralized information and improved resource management. Unfortunately, from a technical point of view, some lack of foresight has proved to be hampering when it comes to the system’s growth and development. Because of the nature of information being dealt with, issues such as security and transactional integrity were never really looked at during implementation. Furthermore, since it was to be employed internally, the need to protect intellectual property was never considered. Finally, when it came to designing the interface, focus was put on simplicity rather than a consistent, functional and intuitive look and feel. When examined from this perspective, it is clear that the current system lifespan is fairly limited and that a major revamp would be required if any kind of commercialisation was being considered.
REFERENCES
Anon . (n.d.) InfoWorld Survey URL:
Anon. (2003) Internet User Forecast by Country URL:

COPYRIGHT
Francois Latouche and Peter Busch © 2005. The authors assign to ACIS and educational and non-profit institutions a non-exclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive licence to ACIS to publish this document in full in the Conference Papers and Proceedings. Those documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. Any other usage is prohibited without the express permission of the authors.