

Web-Enabling Software for Real-Time Online Automated Services

Ronald J. Glotzbach, Carlos Morales
Purdue University

Abstract

The Web has seen its share of trends come and go, but now it is evolving into a centrally located collaboration tool for all professions. The focus is shifting to harnessing the Web's usefulness to gain a competitive advantage in the marketplace. This paper assesses the purposes, usefulness, advantages, and disadvantages of web-enabling desktop application-based software packages. Web-enabling software packages allows users from around the world to perform operations from any location that previously needed a user sitting at a specific computer running specific software applications. It also provides a means to perform operations on the server completely unannounced to the user. The necessary steps for web-enabling software are outlined and explained in this paper. The versatility that this concept offers provides businesses and academia alike with a realm of possibilities that would increase productivity, save time, and save money.

Introduction

The cloud is where ideas come to life. Boundaries are limitless for this centrally located collaboration tool, which happens to be neither central to any one point, nor located in any particular spot. Anything may be brought to fruition in the cloud, the Internet that is. Web-enabling software has been around for a while, but it is more recently that companies have begun to exploit its capabilities. The purpose, usefulness, and advantages of using a web-enabled system will become readily apparent as an example of how it can change the way a company performs its day-to-day business is presented. Simplifying a process by automating it using the Web can help reduce the time commitment by parties involved, reduce ambiguity, improve communication and collaboration, and has the potential to save a company money in the long term. On the other hand, implementing a system can be costly and may extend the project deadline or not be in place in time for its utilization. For those companies with the need and capabilities to utilize it, web automation can be a valuable asset as it opens the door for amazing potential that the software alone did not possess³.

Purpose

The purpose of web-enabling software is twofold, to provide a location that is central to all parties involved and also simplify a process. Utilizing the Web as a collaboration tool makes it easy to simplify a process. Any person, from any place, can access the Web, which makes it perfect for companies that either have vendors, partners, clients, or just separate locations to work from⁴. The Web has proven its stature as a collaboration tool with a multitude of chat rooms and discussion boards.

A process such as reengineering the structure, capacity, and design of a grain storage bin could take months to complete. Engineers could be generating drawings in a CAD package, emailing those files to a partner company a thousand miles away for verification and structural analysis, while attempting to communicate with vendors of the materials what changes they can expect over the next year.

This process can be further complicated by communication breakdown between two or more of the parties involved. The specifications may not clearly define what parameters need to be met. Incorporation of a safety door near the bottom of the bin to evacuate grain in case of accidental submergence into the grain may not have taken into account the additional strength required by the rest of the structure in order to keep the bin from buckling. The process of finding this error, reporting it back to the engineer in charge, and obtaining new specifications for the project may take days or weeks due to the promptness of the email being sent, the call being made, or possibly not communicating at all.

Automating this process on the Web, where all parties can readily participate, would reduce the amount of time needed to bring this project to completion. When the engineer uploads the files to the server, an email notification is automatically sent to their partner who can immediately begin verification and analysis. This functionality can take place on the Web; as the partner is running tests and making adjustments, the engineer can view the results of the tests and see how they are progressing. If there is a major error, as discussed above, by flagging it within the automated system it will automatically send an email notification back to the engineer for immediate attention.

The purpose is seen in this example as both providing a central location, the Web, and simplifying the existing process. Email still plays a role in the process, but it has been automated, taking the responsibility out of the user's hands and sending immediate information to others involved.

Usefulness

Its usefulness stems from the purpose of web-enabling software, which will also lead into the advantages for taking such an approach. In the previous example, it's clear to see what usefulness it can have. Communication between the two parties became simpler as a direct result of the automation.

The usefulness of web-enabling applications can be further seen by considering cases in which developers need to integrate heterogeneous components running on separate machines. By web-enabling an application, the developer gains the ability to make remote procedure call (RPC) to any other web-enabled application. The net result is a system capable of scalability and flexibility. It is scalable because processing intensive portions of an application can be deployed on more powerful machines and flexible because the component of the web application can be changed at any time.

Advantages

One of the main advantages of web automation of software is improved communication. When a typical email message is sent, if there is not an immediate response, it can take anywhere from

hours to a day or two to get a response back. That is more the nature of email in this day in age; many people choose only to check their email at a certain time of day so as to not allow it to dominate their work time. Spam email is more common as well, causing a loss in desire to check it. Along with that, it is often hard to spare the time to sit down and write and email, prolonging the communication gap. Worst case scenario, on a normal schedule, it could take as long as a week for two people to communicate an idea or concern.

Along with improving communication, it also saves time. Instead of taking days or a week to communicate, the automation takes care of it with an immediate response. The software is also located on the web server, which allows both parties to utilize it and view the results of the others work, real-time, instead of passing that information back and forth via email or other means.

Web-enabling software and automation of the existing process also serves to reduce ambiguity that previously existed. By having the Web serve as a central convergence point for everybody involved, and maintaining a series of steps for the process that must be followed online, any ambiguity regarding the next procedure or future expectations of an individual are reduced significantly since they are readily apparent to all parties and no longer have to be solely conveyed by the project manager.

Another advantage offered is the potential for money to be saved as a result of reduced time commitment, reduced ambiguity, necessity for fewer application licenses, and improved communication¹. Time is money, as the cliché goes, thus any reduced commitment for engineers ultimately results in money saved to either invest in other initiatives or mark towards quarterly earnings.

The last advantages are somewhat obvious, but nonetheless important, automation and collaboration. Automation of the existing process is a positive step towards reduction in costs and confusion. Having the software reside on the server allows collaboration to take place over a distance. The operations that are performed, the tests that are run, and the files that are manipulated are all completed in the cloud using software that operates unannounced to the users of the system. This also allows proprietary collaboration software to remain under company protection while permitting partners and vendors to utilize it.

Disadvantages

The initial investment to get the project underway may be costly in some instances. If no system is in place for an intranet and/or Internet application that allows web-enabled software to exist, then the company may be looking at the acquisition of new or additional hardware to get set up for the project. In most cases, there is likely to be an existing architecture in place that upper management would subsequently require the use of for such a project in order to reduce additional costs.

Another disadvantage would occur when the time to develop a web-enabling solution takes longer than that of the time to complete the original project, making the former negligible. Each case where a software package is web-enabled is different in nature; primarily because each software package was developed by a different company that used a different coding structure, language,

or plug-in support. Some software packages may not allow batch mode operations, direct command line instruction, or internal access to containers and classes needed to make the software execute as desired. Much of that is contingent on the nature of the software and its proprietors.

Steps for Web-Enabling Software

1 Assess the need for web-enabling the software package

For some, a need may not exist to warrant this process. A definite need should be established before the time and money is spent on developing this new tool.

2 Determine the platform for deployment

On UNIX machines, Java may be the language of choice, versus C++ or other languages that would likely be chosen on a Windows platform.

3 Research existing solutions to the software need

Reinvention of the wheel is never fun. If a product exists on the market for a reasonable price, consider using it, or at least modeling it.

4 Evaluate potential server-side technologies

Decisions made here can affect the rest of the project. Solutions such as JSP, .NET, services versus applications, and database management systems like Oracle, MySQL, and SQL Server need to be evaluated.

5 Determine if the software supplies access to source code

Some software packages openly allow access to create plug-ins or otherwise access the source code. Having open access provides a means to web-enable without requiring permission or otherwise cracking the software.

6 Define the site specifications to be followed²

Decisions about tasks to be performed such as levels of access, account types, functionality, and specifications for the site layout and design should be made before moving on.

7 Choose a web development software package for layout/design

Some of the most popular packages include Macromedia Dreamweaver, Microsoft Visual Studio, and IBM WebSphere. The choice is typically up to the designer, depending on their level of expertise.

8 Develop the code to interact with the software

Using the above access to the code in conjunction with the chosen server-side technology, interaction code should be written to conduct a 'Hello World' example for functionality and then expanded to obtain the end product.

9 Combine technologies to join the web front end with the automated back end

In some cases, middleware may need to be developed here, but if designed

correctly, the front-end should ‘snap-on’ to the back-end without much hassle.

10 Move the site to a testing server for extensive testing

Begin by testing the application. Find bugs and catalog them for later development.

11 Refine and optimize the code to improve time delays

Based on the test findings, fixes may need to be implemented. This is also a prime time to optimize code that was initially written as a work-around.

12 Move to a production server for deployment

After extensive testing and refinement, move the stable application to the production server. Additional testing can be done from this point; however the environment as a whole should be stabilized.

Conclusion

Web-enabling software provides a way for companies to gain a competitive advantage. Placing software in the cloud and building a collaborative environment around it is becoming a common way of harnessing the power of the Web. Although the initial investment could potentially be costly and time-consuming, following these basic steps for web-enabling software will offer a central location for the automation of the existing process which will lead to reduced time commitment, reduced ambiguity, and long-term monetary savings².

Bibliography

[1] *The new era in enterprise computing*. (2002). Retrieved January 13, 2003, from <http://www.openconnect.com/>

[2] *Web & E-biz*. (2002). Retrieved January 13, 2003, from <http://www.synergex.com/solutions/web/solutions/WebEnableSteps.asp>

[3] *Web enable your products*. (2002). Retrieved January 13, 2003, from <http://www.web-enable.com/>

[4] *Why web enable?* (2002). Retrieved January 13, 2003, from <http://www.resqnet.com/>

RONALD J. GLOTZBACH

Ronald is a Visiting Assistant Professor for the Interactive Multimedia Development area in the Department of Computer Graphics Technology at Purdue University’s West Lafayette campus. He earned his B.S. in Computer Graphics Technology and holds a M.S. in Technology. Related interests include dynamic content delivery methods & integration of varying media into highly technological solutions. Contact him at: rjglotzbach@tech.purdue.edu

CARLOS R. MORALES

Carlos R. Morales is an Assistant Professor of computer graphics at Purdue University. He holds a BA in Telecommunications and an MS Ed. in Curriculum and Instruction. Prior to working at Purdue University, Carlos worked as a Technical Director. His research interest includes distance learning, animation, and multimedia development. He can be reached at crmoraes@tech.purdue.edu