A SOAP-Enabled System for an Online Library Service

Frank L. Walker (walker@nlm.nih.gov)
and George R. Thoma (thoma@nlm.nih.gov)
National Library of Medicine

Keywords: DocMorph Server, SOAP, Web Service, Image Processing, World Wide Web, TIFF, PDF, DocView, NLM, Internet

Abstract: This paper describes an application of the Simple Object Access Protocol (SOAP) technology to improve the performance of a prototype web-enabled system, DocMorph, developed at the Lister Hill National Center for Biomedical Communications, an R&D division of the National Library of Medicine. DocMorph provides online information processing such as file format conversion (e.g., from about 50 file formats to PDF), extraction of text from image files, and the conversion of document images or word processing files to speech using a combination of OCR and speech synthesis.

In the 2 3/4 years of its operation, many of DocMorph's more than 4,000 registered users have submitted more than 56,000 jobs to the system. This amounts to 55 GB of data, or about 600,000 document pages. Most submissions are for the conversion of TIFF images to PDF, enabling platform-independent document delivery and easier usage of the documents.

While DocMorph serves as a useful tool for a user community, we are using it as a research and development test bed to seek better ways of processing library information. One area investigated was improved performance for users submitting multiple files at a time for conversion to PDF. SOAP promises to be an effective technology for this purpose because it enables the design of a client software program called MyMorph that allows a more efficient conversion process than submitting files via a web browser. The integration of SOAP into DocMorph gives users the option of accessing it either via MyMorph or via web browsers.

1. Introduction

Since the early 1980's document delivery by libraries and information service providers has evolved from use of the postal service to delivery via the Internet. With the widespread use of the Ariel™ system developed and distributed by Research Libraries Group, several thousand libraries have used it to conduct interlibrary loan services electronically via the Internet.[1,2] In addition, with client software such as DocView (developed at NLM's Lister Hill National Center for Biomedical Communications), library patrons in the late 1990's began benefiting from Internet document delivery.[3,4] DocView runs on all Windows™ operating systems, and enables a library patron's desktop computer to receive documents sent by a library's Ariel system. DocView is

capable of displaying monochrome bitmapped images in either the Group on Electronic Document Interchange[5] (GEDI) file format used by Ariel systems, or in the Tagged Image File Format[6] (TIFF). More than 11,000 users from 160 countries have registered to use DocView.

As an adjunct to the DocView project we have developed a web-mediated resource called the DocMorph Server to enable the investigation of issues of delivering, processing and using electronic library information.[7] This is part of an ongoing R&D program in document imaging that has spanned many aspects of electronic document conversion and preservation, Internet document transmission and document usage. The DocMorph Server, located at http://docmorph.nlm.nih.gov/docmorph, was launched in May 1999. It enables remote users with web browsers to upload files for processing in any of five ways. First, it can convert files in any of more than fifty format types to the Portable Document Format (PDF™). The types of files include black and white images, grayscale and color images, and word processing files. Second, it can convert any of these file types to TIFF images. While PDF has the advantage over TIFF in portability, TIFF images are easier to edit, since PDF readers do not allow editing. To help with TIFF editing, DocMorph includes a third function for splitting a multipage TIFF file into single TIFF images. DocMorph also includes a fourth function for extracting text from any of the file types it processes: in 1999 it became the first publicly available web site to offer image-to-text conversion via optical character recognition. Finally, as a tool for researching and improving accessibility to library information, DocMorph has a function for changing files to synthesized speech.

While DocMorph serves as an R&D tool to investigate the utility, reliability and speed of image and information processing algorithms, it also serves a user community by providing the functions mentioned above. In its first 2 3/4 years of operation, more than 4,000 people registered to use DocMorph. They submitted to the system more than 56,000 jobs consisting of 55 GB of data, or about 600,000 document pages. Of the total number of jobs submitted, over 95% were processed successfully. Files that could not be processed consisted of file types not handled by DocMorph, and defective files. Usage statistics gathered during this time are given in Table 1.

DocMorph Function	Jobs Processed Successfully	Percentage of All Jobs
Convert Files to PDF	40, 459	75.8%
Convert Files to TIFF	1,145	2.2
Split TIFF Images	2,049	3.9
Extract Text	6,898	12.8
Synthesized Speech	2,856	5.3

Table 1. DocMorph Function Usage May 1999 – February 2002

Many libraries around the world are using DocMorph as part of their daily operations, especially for converting scanned images to PDF. Once a document delivery librarian has the file in PDF format, it may be delivered to patrons as an email attachment. The

job processing time becomes a factor if the librarian needs to submit a number of jobs to DocMorph. The average times for DocMorph to process jobs for each of its five functions are given in Table 2.

DocMorph Function	Average Time per Job (seconds)
Convert Files to PDF	12
Convert Files to TIFF	9
Split TIFF Images	10
Extract Text	17
Synthesized Speech	12

Table 2. Average Job Processing Time

As shown in Table 2, DocMorph usually takes less than 20 seconds to process a job. However, the times shown do not include the time taken by the user to select a file from hard disk, upload it to DocMorph, download the results back to the user's computer, and choose a folder on the hard disk to save the results. These operations may take far more time than it takes DocMorph to process the job. Depending on the user's Internet connection, the time taken for uploading and downloading files can vary considerably. A typical ten-page file consisting of scanned black and white document images is about one megabyte in size. If the user has a high speed Internet connection, e.g., 1 Mbs, uploading and downloading may each take less than twenty seconds. However, if the user has a slow modem connection at 33 Kbs, it will take several minutes each for uploading and downloading. Hence, using a web browser for processing files via DocMorph can take a considerable amount of operator time, especially if the user has a slow Internet connection, and has several files to process. The question is: How can this be improved?

The solution pursued is to design client software that replaces the browser and provides a significantly improved user interface that eliminates the drawbacks of the browser interface. It allows the user to select multiple files on the local computer for submission to DocMorph. The client software handles all file uploading, waiting for job completion, downloading and storing the results on the computer hard disk. In short, it eliminates the operator attention required by allowing the operator to do other work while waiting for the client and DocMorph to process the jobs. We have designed this client program with a new technology called Simple Object Access Protocol (SOAP), and augmented DocMorph to use it. At this point, the client program, called MyMorph, is designed to handle only one of the five functions provided by DocMorph: conversion of files to PDF, since this is DocMorph's most-used function, and it provides an excellent means for testing and evaluating the new functionality.

2. SOAP: Simple Object Access Protocol

SOAP is a new technology that promises to provide services via the Internet, called Web Services. It is an Internet protocol that allows a pair of computers, typically a client and a server, to communicate with each other. The information exchanged uses Extensible

Markup Language (XML) commonly sent via Hypertext Transfer Protocol (HTTP), although it can also be sent by other communication protocols such as File Transfer Protocol (FTP) or Simple Mail Transfer Protocol (SMTP). SOAP offers advantages over competing methods of providing web services, such as Microsoft's Distributed Component Object Model (DCOM) and CORBA's Internet Inter-ORB Protocol (IIOP). These latter techniques provide faster communication than SOAP because the information is typically sent in the form of smaller messages. However, SOAP's chief advantage over DCOM and IIOP is that it can be sent via HTTP, which is used for all web communication. While firewalls will almost always block DCOM and IIOP, they seldom block HTTP. A good way to provide Internet services, and not be blocked by firewalls (common in most organizations) is to use an HTTP-based service such as that provided by SOAP.

We used Microsoft's SOAP toolkit version 2 to implement MyMorph and necessary modifications to DocMorph. For this application, SOAP consists of an XML message exchanged via HTTP between MyMorph and DocMorph. A SOAP message as shown in Figure 1 consists of three parts:

- The Envelope, which is a top-level container for the message
- The Header, which contains added features for the SOAP message, and
- The Body, which contains information for the recipient of the message.

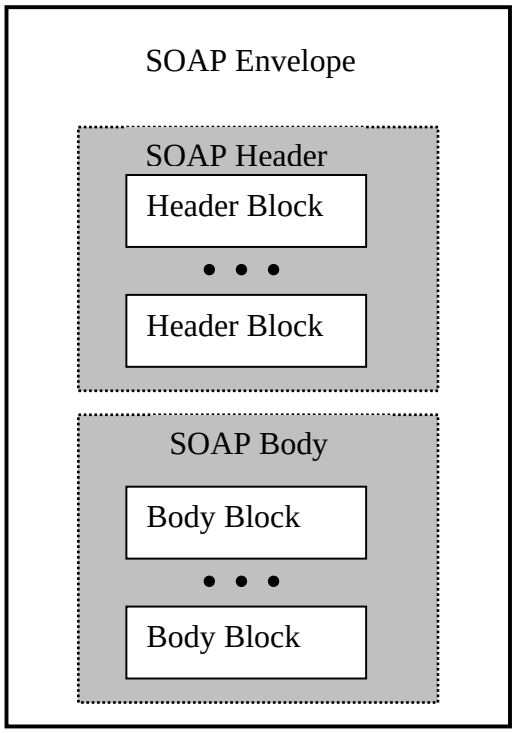

Figure 1. SOAP Message

The following is a typical SOAP message sent from DocMorph to MyMorph. This example shows the envelope with a body but without the optional header. This is the message sent for MyMorph's GetVersion function, which allows the client to determine whether there is an updated client version available from the DocMorph Server. The body parameter Result gives the latest version number available, and WebAddress gives the location on the web where the latest version may be obtained.

```
<xml version="1.0" encoding="UTF-8 " standalone="no "?>
<SOAP-ENV:Envelope SOAP-
ENV:encodingStyle=http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP-
ENV=http://schemas.xmlsoap.org/soap/envelope/>
  <SOAP-ENV:Body>
        <SOAPSDK1:GetVersionResponse
        xmlns:SOAPSDK1=http://tempuri.org/message/>
        <Result>1.0></Result>
        <WebAddress>http://docmorph.nlm.nih.gov/mymorph/setup.exe>
        </WebAddress>
        </SOAPSDK1:GetVersionResponse>
   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3. Modifications to the DocMorph System

DocMorph was originally designed to handle user requests from web browsers. Its multi-computer architecture permits expansion up to ten processors. One processor, termed the Permission computer, contains a Microsoft Access database that keeps track of all transactions throughout the system, and retains this information for the long term. The remaining computers, of which there may be a maximum of nine, are called the Worker computers. Each of these has a short-term database that keeps track of the transactions on that specific machine. Web browsers are directed to the Permission computer to request a particular job, such as file conversion to PDF. In response to such a request, the Permission computer examines the entire system to first determine which of the Worker computers can handle that type of job. Of those computers, it selects the one that has the least amount of current and pending work. Then it routes the browser request to that computer, at which the user's file is processed. This job switching mechanism ensures that jobs are distributed evenly among the Worker computers, so that no one computer is overloaded.

To accommodate SOAP-based MyMorph clients while at the same time handling web browsers, modifications were made to DocMorph's Permission and Worker computers, as illustrated in Figure 2.

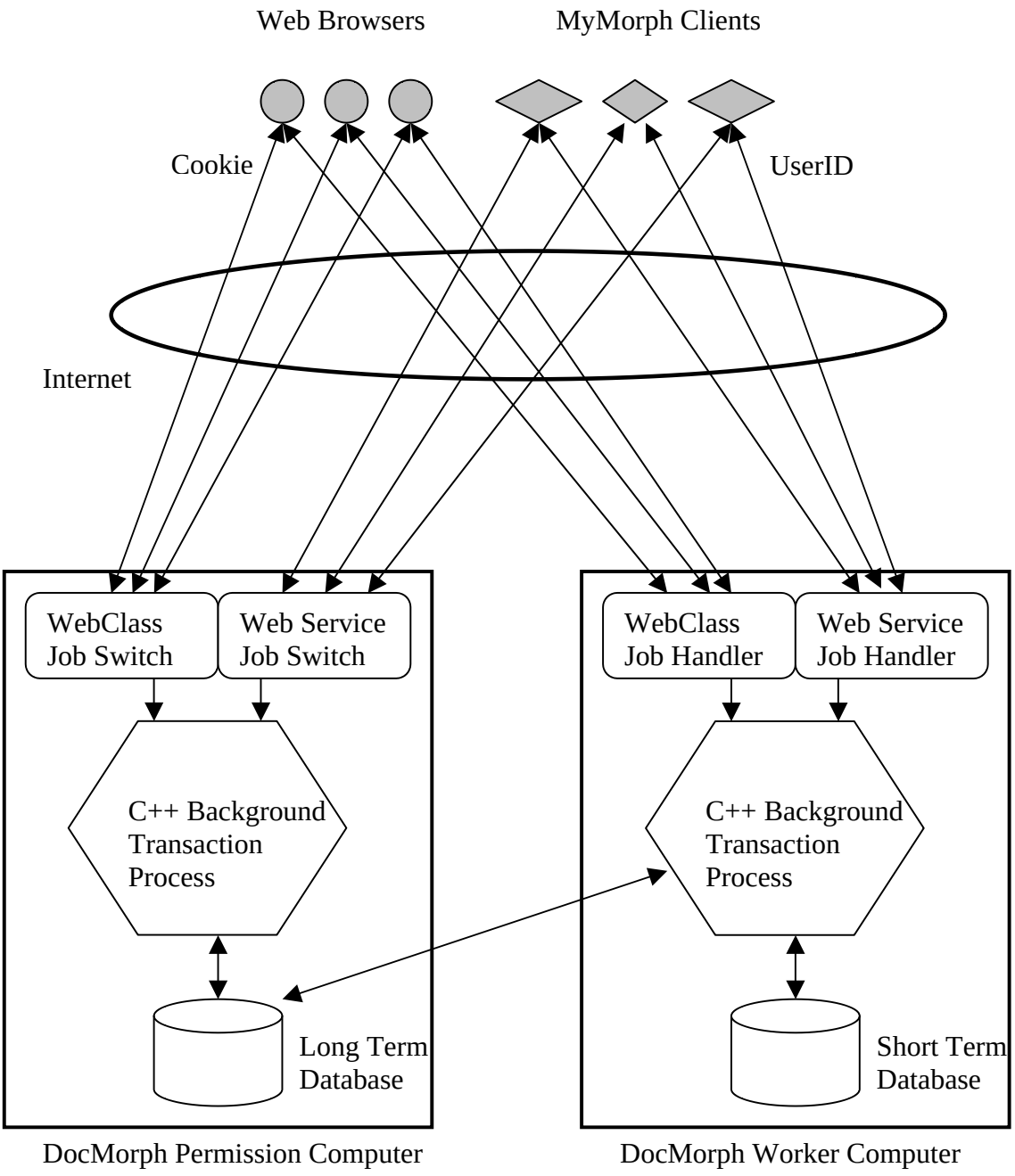

Figure 2. DocMorph System Architecture

DocMorph communicates with the outside world, whether with web browsers or MyMorph clients, through dynamic link libraries (DLL's) written in Microsoft Visual Basic. In the case of web browsers, the DLL's are WebClass objects that are instantiated when invoked by the browser. For communications through MyMorph clients, the DLL's are Web Service objects that are instantiated and run. In each case, either the WebClass Job Switch or the SOAP Web Service Job Switch at the Permission computer decides which Worker computer will receive the job. The Worker computer's WebClass and Web Service Job Handler objects handle the file uploading and downloading. A C++ module does the actual background processing and transaction handling in each machine. To keep track of MyMorph users, instead of using a cookie the system uses a 32-character UserID assigned to each MyMorph user on initial registration. The MyMorph client sends the UserID each time it requests a service, and DocMorph uses this UserID to keep track of system utilization.

4. MyMorph User Interface

The MyMorph client is a Windows application created by using Microsoft Visual C++ and the Microsoft SOAP toolkit. It allows the user to select multiple files for conversion to PDF via DocMorph. The user interface consists of a single dialog box containing two windows (Figure 3).

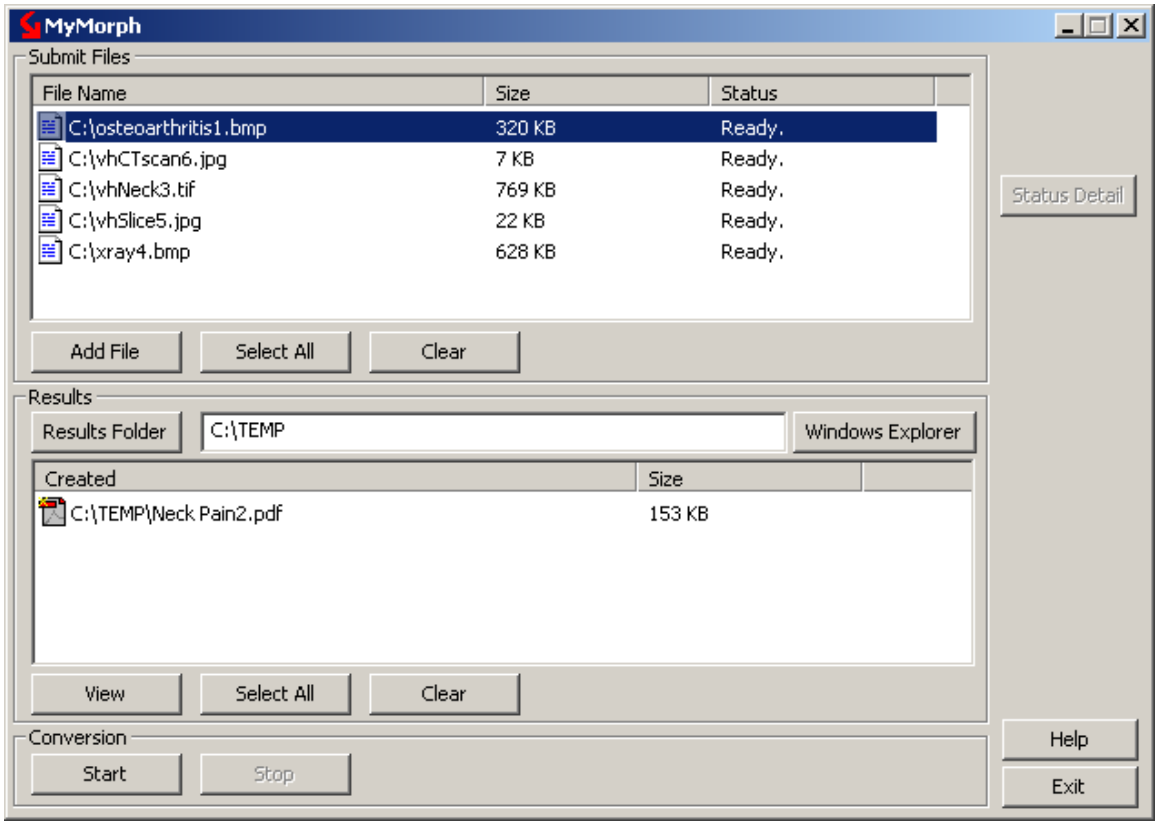

Figure 3. MyMorph User Interface

The top window, called Submit Files, lists all files the user has chosen to convert to PDF. The bottom window, called Results, lists the PDF files returned to the user. The Add File button invokes a dialog box for selecting files from one of the computer's disks. The user may select multiple files simultaneously using this dialog box. Once the files for PDF conversion are chosen, the user simply clicks the Start button. From then on, MyMorph handles the entire process: uploading each file to DocMorph via the Internet, waiting for job completion, downloading the result, and storing the PDF file on the user's hard disk. These steps that previously took considerable operator time are now transparent to the user until all files have been converted. The major advantage of MyMorph is that it allows the conversion of several hundred files at a time in this manner, with the only effort required by the user being to initially select the files for conversion. The MyMorph user interface also has a built-in help facility (Help button), a function for displaying the PDF files using Adobe Acrobat Reader (View button), a function for displaying error status (Status Detail), and a Windows Explorer button for managing the files in the Results folder.

5. MyMorph SOAP Communication

The MyMorph client has two channels of SOAP communication with DocMorph. The first is communication with the DocMorph Permission computer, and the second is with any of the Worker computers. The MyMorph client initiates the communication by downloading a Web Services Description Language (WSDL) file from the computer. The WSDL file uses XML to describe the service provided by the server.[9] For DocMorph, the Permission computer's WSDL file defines the protocols for three main functions: Register, GetVersion and GetPermission. The first time MyMorph runs on the user's computer, it displays a dialog box that allows the user to register to use the software. MyMorph uses the Register function only once to send the information to DocMorph. In response, DocMorph sends back a 32-character UserID that is saved on the MyMorph computer in the system registry. MyMorph uses the GetVersion function upon startup to obtain the latest version number of the MyMorph client software. If there is a newer version available, the MyMorph software allows the user to download and install it. This function also returns the web address at which the latest MyMorph version is located.

The third function available at the Permission computer is GetPermission. MyMorph sends the GetPermission request to the Permission computer for each file to be converted. Included with each request is the 32-character UserID that is assigned to each user upon registration. The status that DocMorph returns for this function contains two pieces of information. The first is an indication as to whether the MyMorph client can immediately send a file for processing. If DocMorph is busy and cannot handle MyMorph's request, this status data indicates a time delay before the client may ask for permission again. The second item of information is the path to the Worker computer to which the client is routed.

Once MyMorph is routed to an appropriate DocMorph Worker computer, it initializes its communication with that computer by downloading its WSDL file. This file describes

the protocol for file conversion to PDF, and there is one function available here: MakePDF. For each MakePDF request sent to the Worker computer, MyMorph sends its UserID along with the file to be converted. All of this communication takes place in an XML-formatted message. The response to MakePDF contains the resulting PDF file, also embedded in an XML-formatted message.

6. Current Status and Test Results

The MyMorph client and modified DocMorph server are undergoing extensive in-house alpha testing prior to release for outside beta testing. Initial results are highly encouraging, and minor bugs are being found and eliminated in the client and server software. So far, the only disadvantage found in the current SOAP implementation has been in its handling of binary files: the only method the current toolkit offers for transmitting binary files is via base64 encoding. This means that any file that is sent from one computer to another must be base64 encoded. Our tests have shown that this coding method increases the size of typical image files by one-third. So, for a typical ten-page journal article scanned at 300 dots per inch resolution that resides in a 1-megabyte file, the encoded result is about 1.3 megabytes in size. This results in a greater bandwidth requirement when sending the file, and creates additional overhead at the sending computer for encoding, and at the receiving computer for decoding. A better method of binary file transmission is required, and indications are that future SOAP implementations may include other forms of file attachment such as Multipurpose Internet Mail Extensions (MIME) or Direct Internet Message Encapsulation (DIME).[10]

A simple test demonstrates the potential utility of SOAP and the advantage of MyMorph over the existing DocMorph web-based user interface. We compared the times to convert five 970-kilobyte TIFF files to PDF, with both MyMorph and DocMorph interfaces. We ran the tests at two different communication speeds: 100 Mbs direct-connect Ethernet and 26.4 kbs dialup modem, since most users are likely to encounter speeds between these extremes. As Table 3 shows, regardless of the communication speed, it takes less time to process the five files using MyMorph than with the web-based DocMorph interface. As noted previously, the speed of MyMorph will increase once binary transmission of files is included in the SOAP toolkit.

	100 Megabits/sec	26.4 Kilobits/sec
DocMorph Web Interface	3 min 2 sec	1 hour 51 min 46 sec
MyMorph	35 sec	1 hour 5 min 58 sec

Table 3. TIFF to PDF Conversion Times for Five 970 Kb Files

An important point to be noted in Table 4 is the amount of time required by the operator for these five conversions. In contrast to the considerable amount of operator time spent in monitoring and interacting with the web-based interface, the MyMorph interface requires only 10 seconds of operator time.

	100 Megabits/sec	26.4 Kilobits/sec
DocMorph Web Interface	3 min 2 sec	1 hour 51 min 46 sec
MyMorph	10 sec	10 sec

Table 4. Operator Time Required for Converting Five 970 Kb Files

7. Summary

This paper has described the integration of a new technology, SOAP, into an existing prototype web-based service (DocMorph) that provides document image and information processing. This development allows the web site to provide two methods of access. The first, a conventional method, involves the use of a web browser for uploading files to the web site for processing. The second method, using SOAP, is implemented through a unique client program that significantly reduces the human interaction required by a web browser. For the case where multiple files are to be processed by the web site, the SOAP-based method is expected to increase user productivity.

8. References

1. Berger, M.A., "Ariel Document Delivery and the Small Academic Library," College & Undergraduate Libraries, Vol. 3(2). The Haworth Press, 1996; 49-56.
2. The World Wide Web address for Research Libraries Group is located at this URL: http://www.rlg.org.
3. Walker FL, Thoma GR, "DocView: Providing Access to Printed Literature through the Internet," Proceedings IOLS'95. Medford NJ: Learned Information, 1995; 165-173.
4. Walker FL, Thoma GR, "Internet Document Access and Delivery," Proceedings IOLS'96. Medford NJ: Learned Information, 1996; 107-116.
5. Information on the Group on Electronic Document Interchange (GEDI) format is available at URL http://lib-www.uia.ac.be/MAN/T02/t51.html.
6. TIFF Revision 6.0, Aldus Corporation, June 3, 1992.
7. Walker, FL and Thoma, GR, "Web-based document image processing," Proceedings of IS&T/SPIE Conference on Internet Imaging, San Jose, California, January 2000, 268-277.
8. Walker, FL and Thoma, GR, "Read It To Me!", Proceedings of the Twenty First National Online Meeting, May 16-18, 2000. Medford NJ: Information Today, 2000; 473-483.
9. Kreger, H. "Web Services Conceptual Architecture (WSCA 1.0)." May 2001: IBM Software Group. International Business Machines Corporation, Somers, New York.
10. Powell, M. "DIME: Sending Binary Data with Your SOAP Messages." Microsoft Corporation, January 2002. Available at: http://msdn.microsoft.com .