

Profiting From Research Data

Addressing Web Service Applications for Proprietary Data

Celeritas Technologies LLC

November 1, 2002

Prepared By:

Celeritas Technologies, LLC

7101 College Blvd, Sixth Floor

Overland Park, KS 66210

(913)491-9000



Copyright © 2004 Celeritas Technologies, L.L.C.

This publication may not be reproduced, stored in a retrieval system, or transmitted in whole or part, in any form or by any means, electronic, mechanical, audio, photocopying, recording, or otherwise, without the prior written permission of Celeritas Technologies.

While this information is presented in good faith and believed to be accurate, Celeritas Technologies does not guarantee satisfactory results from reliance upon such information.

Celeritas Technologies reserves the right, without notice, to alter or improve the designs or specifications of the products described herein.

Nothing herein is to be construed as a warranty or guarantee, expressed or implied, regarding the performance, merchantability, fitness, or any other matter with respect to the products, nor as a recommendation to use any product or process in conflict with any patent.

All products, languages, or trademarked names that are mentioned in this document are acknowledged the proprietary property of the respective owner.

Executive Summary

Publishing companies often create publications that contain vast amounts of data. Customers of these publishing companies place value on that data, and purchase it in traditional forms such as print or on the Web. However, these traditional forms of gaining access to data do not always meet the needs of the customer; for example, the customer who would like to gain *system level access to the data*.

This paper discusses a means by which a publishing company can provide an effective, flexible and scalable interface to their customers' systems, allowing those customers access to the data that is of value to them or their system without the limitations of traditional mediums such as print or a web page. There are a number of ways such an interface could generate significant revenue for the publishing company, and be resold to other customers with similar needs.

The Value of Data

Publishing companies often have a number of publications that require a great deal of research to produce. Examples include reference publications such as vehicle or equipment pricing guides. The publishing company's research staff must accumulate a vast amount of data in order to produce these guides. And, in the case where these guides are periodicals, that research has to be repeated time and time again. The research department represents a competitive advantage for the publishing company: customers usually find it less expensive to purchase the data they need in the form of publications than to try to gather the data by their own means.

So how can a publisher *profit from this data*? Obviously, the published media is one way. Another way is by providing the data directly to a customer. For example, if an insurance company wanted to use vehicle pricing data within their own organization, on their own computer systems, the publisher could provide the data electronically, for a price. A third approach is for the publisher to make the data available on a web site, and then sell subscriptions to access the web site.

The point of this paper is that the **data** in your research-driven publications **has value**.

Is your publishing company reaping the profit potential of the data that it owns? It is likely that that data has value to large organizations such as financial institutions, government agencies and others. These large organizations represent a potential for greater profitability.

This paper discusses a technology solution that can provide another source of revenue for the data you already collect in order to publish reference media. This solution could be implemented by any publication that owns data that is needed by other organizations. By using new approaches to the distribution of that data, there is great potential for you to increase your profit margins.

Traditional Distribution Approaches

There are a number of traditional approaches publishing companies use to distribute their products. As an example, we will look at a company that produces reference publications used for pricing vehicles. Historically, this publishing company primarily sells the data it has accumulated via research through pricing books. This approach has worked well in the past and will not disappear any time soon.

The company may also distribute data via electronic file transfer. Such a file transfer can be as simple as attaching an Excel spreadsheet to an email message, or it can be accomplished using file transfer protocol to transfer files directly to the customer's computers.

The World Wide Web is yet another way for reference data to be made available. Most publishing companies today have some kind of online presence, and many are publishing their data via the web (e.g., the Kelly Blue Book, or NADA, and financial publications such as MorningStar and ValueLine). Publications like these are currently generating revenue from the web, either through advertisements or site subscriptions or both. Publishing data on the web can provide rapid update capability, as well as a flexibility and ease of use for customers that is simply not available in printed material.

New Opportunity

This paper came about in part as a result of watching the interaction between a publishing company and a financial institution that was one of its largest clients.

Among other things, the financial institution was involved in insuring construction equipment. This part of their business was the loss leader for their insurance operations. They believe that the major cause of their losses was inaccurate valuation of equipment. They were insuring equipment at incorrect values.

The publishing company publishes a pricing guide that has become an accepted standard in the equipment valuation industry. The financial institution wanted to use valuations from the pricing guide in their own computer system to enable them to correctly value equipment. While the pricing guide was available in printed form, as electronic files, and on the web, all of those methods of distribution fell short of the ideal solution.

Disadvantages of Current Distribution Methods

A printed pricing guide would have required the financial institution to engage underwriters to manually check the valuations. This was deemed to be impractical for the volume of valuations that had to be made, and was also error prone.

It was also determined to be too time consuming for underwriters to look up every piece of equipment on the web site.

Even the possibility of electronic file transfers fell short of the ideal solution, as they were inflexible, would happen only periodically, and required a new support structure to

accommodate them. Also, the system lacked the ability for seamless synchronization with the publisher.

The Direct Interface Approach

The solution is to provide a means by which the financial institution could directly query the data they needed *from the publisher's system*. The direct query would be done through an interface or a remote procedure on the client's system that the publisher would make available. The remote procedure would be as easy to use as any other procedure on the client's system.

This interface would give the financial institution's existing system access to a particular valuation data set from the publisher. This data might be valuations for a select set of equipment. The remote system calls to the publisher's system could be made from the financial institution as often as different equipment valuations were needed.

For the publishing company this direct interface approach could provide another source of revenue that would have the possibility of creating rather large profit margins. Access to the interface could be fee based. This fee could be similar to a subscription or might be based on usage (the financial institution would be charged a fee based on each call to the interface). The same direct interface could then be sold to other financial institutions, government agencies or to other customers who needed ready access to the data.

Introduction to Web Services

This "direct interface" idea is nothing new. It is known as **distributed computing**. What *is* new, however, is the *manner* in which it can be implemented and the apparent ready adoption of this technology in the current marketplace.

There has been a great deal of "hype" surrounding XML and web services, yet there is truth in all that hype. XML is simply a way to represent data. Web services are, in essence, a way to provide an external interface to a system. In the example above, the publishing company could provide external interfaces to any internal system that has valuable data to outside entities' systems via web services.

Web services can be thought of as a remote procedure call using http and XML, two standard web technologies. With a web service approach, external clients access a web service that resides on the publisher's application server. A web service has all the advantages of a web page in terms of portability and accessibility. The publisher could define any number of web services that would provide access to the data desired by an external client. Clients would then be able to request information from one or more of these web services depending on the clients' particular needs.

All clients' requests of the web service would be a programmatic. For the client developer it would be much like coding a method call except the method would be a web service owned by the publisher rather than a method local to the client program.

We can use our previous example here. A client application at a financial institution needs equipment valuation data to display to an underwriter. The financial institution's client application would allow for the entry of "Make" and "Model". This entry could be for one piece of equipment or possibly for a list of equipment.

When the nature of the request is determined, the client would call—via Simple Object Access Protocol (SOAP)—the web service residing on the publisher's application server. The marshalling of this list into parameters of an XML SOAP procedure call would be transparent to the client. The web service procedure call would be like any other procedure call in the application. The publisher's application server would take that request and compile the desired results. The results could then be returned to the client as virtually anything, from a raw XML string to a list of equipment value objects.

The client application would then manipulate the data provided by the web service procedure into a format to display to the user or it might use the results to validate a list of equipment valuations.

Web Services in the Real World

Web services is a newer technology but one that will gain a significant foothold in the industry. Microsoft has adopted web services in virtually all of their new applications. SOAP web services are the way in which MS distributed applications will communicate.

The major Java application server vendors are also touting web service building tools. Tools have been developed for most development frameworks to generate web service source for the developer. These tools are available now and make developing web services relatively easy.

There are a number of example web services in existence. The website <http://www.google.com> is a leading search engine site used by millions to query World Wide Web content. Google has a web service available that allows Google queries from inside a custom application. That means that developers can leverage the power of a market-leading web search engine from within their own application. More information is available on this Google interface at <http://www.google.com/apis/>. To gain an understanding of some other uses for web services, a list of web services available for general use can be found at <http://www.xmethods.org>.

Advantages of Web Services Approach

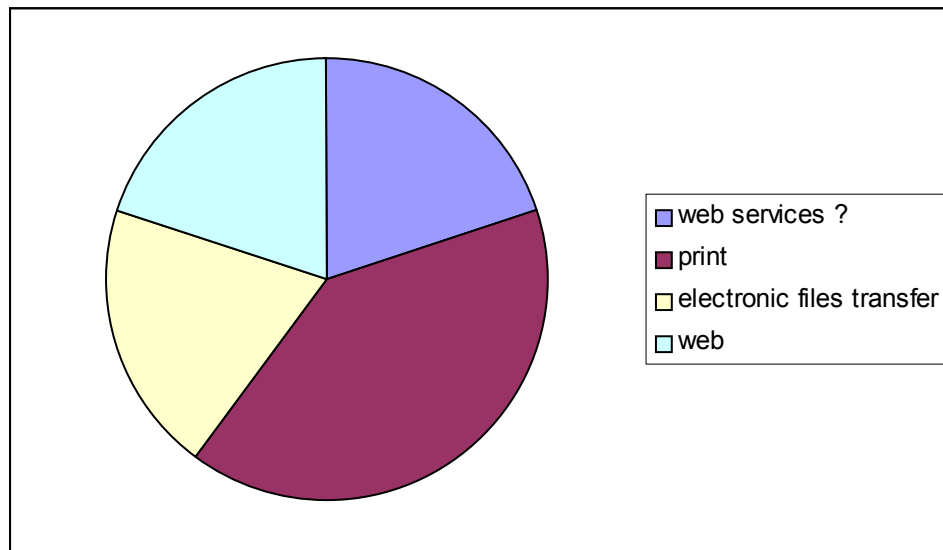
Several advantages exist with a web services approach. Web services is web-based, so accessibility, security and portability are similar to that of other web technologies such as a web page.

Web services are flexible, easy to maintain and reusable. A publishing company would only be responsible for providing *data* via this service. Reference data collection and maintenance is a task the publishing company is well suited for. Costly development and maintenance of the *user interface to the data* would be left to the client. The publisher would have to spend little on this service and the client would have the flexibility to build

a user interface or a complete system customized to their needs, but driven by the publisher's data.

External clients could dynamically interact with live data owned and maintained by the publisher vs. "stale" data that has been loaded into an external mirror database or transported via a flat file. Implementation would be easy for the client, as no supporting system would be required; they would simply build a SOAP-enabled client. This client would then communicate as the interface with the publisher's web service.

The design for external clients to include the publisher's service would be simple as well. Any design should be as simple as delineating the data that would be sent via parameters to a method call to the web service and the return values with which the web service should respond.



It is important for publishers to look at whether they are missing a piece of the 'reference media market penetration' pie shown above. The potential for the publishing company to gain a new piece of that pie is now clear.

It appears that the profitability of this new "web services" piece will be high. In fact, a publisher who currently has no product that can profit from web services is at risk.

On the other hand, the publisher who penetrates this new market will gain a competitive advantage over other publishing organizations, will help protect their enterprise, and will increase margin by adding revenue streams to work that they already do quite well.