

Exposing Enterprise Data: XBRL GL, Web Services, and Google, Part 1

BY GIANLUCA GARBELLOTTO

What would be the value to your organization if the data in it could flow together, regardless of the source, and be accessed by any application—even a standard search engine? Not only is it possible, but it's no more expensive than the solutions that organizations use to consolidate data today, and it can bring benefits far beyond that of an ERP (Enterprise Resource Planning)

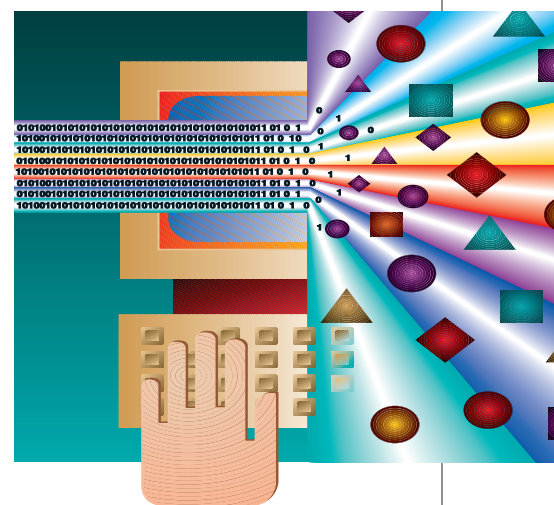
system alone—perhaps even reducing the need for a new ERP system.

In this article, I will discuss the technical and domain standards that can make this solution a possibility. XBRL GL, the standardized Global Ledger, and Web services are the start. Project Nunavut, which will be described next month, is the fulfillment.

At first consideration, many benefits of the eXtensible Business Reporting Language (XBRL) appear to be the same as those of existing, broadly used applications: Centralizing all reporting from disparate modules of the information system, different applications contained within a single enterprise system or a single data warehouse, and eliminating manual reentry and reconciliation of data are

all promises of ERP systems.

This becomes more obvious as the scope of XBRL use is expanded beyond its best known purpose, business reporting, known as XBRL FR. In contrast, XBRL GL, the standardized Global Ledger, is a standard format to represent financial and nonfinancial data at the detail level, move the data between different systems and applications, and provide context for drilling down from summary reporting (XBRL FR) to the detail data that flows to it. This sounds a lot like the value proposition of ETL (Extract, Transform, and Load) applications: interoperability between different, and generally incompatible, systems or modules along with the quest for a common format to represent data no matter



where it comes from or goes to.

Still, XBRL GL is in no way intended as a substitute or replacement for ERP systems or ETL applications. We still need a database or data warehouse to report from and a user interface and applications, just as we need specialized tools to perform translations from different data formats. So what is the real value proposition of XBRL and XBRL GL in particular? XBRL is an open, freely available, and royalty-free standard, an obvious advantage compared to ERP and ETL proprietary solutions.

When software vendors provide XBRL GL as a standard import/export format and the primary payload for Web services, using XBRL

GL will be a transparent process. For now, there are a number of tools for mapping from native file formats into XBRL GL and back. But why go through the pain of mapping source data to XBRL GL and generate instance documents that are much bigger than the source data itself in order to achieve something that's already provided by existing and broadly adopted, though proprietary, applications?

The point is that we need different solutions for new challenges.

No ERP system really lives up to the full meaning of the acronym, even when considering only the traditional goals and functions of this type of application: There's always at least one spreadsheet that represents data not found in the main system or a module that isn't fully integrated for some reason. XBRL GL is an open standard that can help make even the most integrated system more interoperable, and data more reusable, in a cost-effective way.

The real questions that CFOs and CTOs need to ask themselves are:

- ◆ Do we need to share our data with someone who doesn't have access to our ERP system?

- ◆ Do we outsource some functions to external partners who not only need to browse but also to access and edit our corporate data warehouse?

- ◆ Do we have to provide data to analysts, auditors, or regulators?

A closed system such as an ERP application isn't an effective answer to the challenges posed by a distributed environment where data from different systems needs to not only be easily accessible within the entity but constantly exchanged and shared with different parties. In this context, something new and different is needed—a new, standardized layer

of accessibility, interoperability, and reusability of data and processes that integrates the existing information system to make it better serve its traditional purposes and help achieve new ones.

The logical answer is Web services—applications or functions that can be made available through the intranet or the Internet as appropriate, are based on broadly adopted World Wide Web Consortium (W3C) standard protocols and encodings to exchange information (e.g., SOAP and Web Services Description Language—WSDL), and are built on the broad acceptance of XML (Extensible Markup Language) as the universal language for representing and transmitting structured documents and data independent from the programming language, software, and hardware.

But the adoption of Web services and XML obviously isn't enough to ensure that the benefits of standards are fully leveraged. Will my auditor, or the company to which I outsourced the management of my accounts receivables, be able to “understand” the way in which I represent my data with XML? If I describe it or share with them a documented XML schema, the proprietary “dictionary” used within my entity to represent accounts receiv-

ables, they can. But will they bother to understand it, implement it, and use it to represent their own data? Will they do the same with all the other proprietary XML schemas—each one slightly or greatly different from the other—that each of their partners or customers has come up with to represent the same data? The answer is probably no. But even if they do, this is only limited progress from the “closed system” situation described above. A Web-services-oriented architecture makes data available across systems and across entities, but it still doesn't make the data understandable. XML isn't enough. What is needed is an agreement on how to use XML to represent different types of data in a consistent way that is universally understandable.

This is exactly what XBRL Global Ledger is. It is an agreement reached within the XBRL International Consortium on how to use XML and other related technologies (XML namespaces, XML Schema, XLink) to represent data and documents found in operational and accounting system ledgers and subledgers, such as accounting entries, trial balances, charts of accounts, payroll information, AR/AP, inventory items, statistical indicators, and more. Basing a Web-services-oriented architecture on this particular implementation of XML allows building Web services that are truly universal. They can potentially be used by anybody to access, edit, or validate their data, no matter where the data resides and from which application it was generated. Once the data is represented through XBRL GL, it comes with a structure that is always consistent. For example, if the element <entries Type> has a value of “assets,” a Web service built to understand XBRL GL



Answers to puzzle on p. 64.

knows that it is dealing with a fixed assets list and that the main information about each asset can be found in the <measurable> group of elements. Regardless of whether the original fixed assets list comes from QuickBooks, FAS, SAP, a proprietary application, or any combination of those, the information will be presented in a single, consistent, standardized format. I always like to use the expression “Universal Meta-data Structure” to define XBRL GL. In the same way, I believe that XBRL GL-based Web services can be the foundation of a “Universal ERP System.”

Now it becomes possible, for instance, to validate that total debits equal total credits in a batch of journal entries or to extract from a list of accounts receivables those entries that exceed a certain amount or maturity. The only effort required is to map from the legacy system, where the data resides, to the XBRL GL taxonomy, a one-time effort. And the benefits don't end there—that effort grants access to a potentially unlimited set of standardized functions made available by different parties through Web services. This also makes that data available to customers, vendors, auditors, regulators, and business partners with no further transformation, and it virtually eliminates rekeying data and reconciliation burdens. All this can be achieved without replacing a single component of the existing information system.

You can experiment with some live examples of XBRL GL-aware Web services at <http://wixix.net>. You can try them with your own data represented with XBRL GL, or you can use an existing XBRL GL instance document such as those found in GaLaPaGoS (the *Global Ledger Practices Guide for Study*), a

set of “best practice” instance documents created by the XBRL GL Working Group, at <http://gl.iphix.net>. The sample Web services are deliberately very simple, but they provide enough elements to support the approach described in these pages. They were built using standard out-of-the-box .NET technology, with no additional XBRL-specific libraries or functions. The same can be achieved using standard Java technology. In other words, XBRL GL can be used today.

Now we know what XBRL GL and Web services can do for data within your own information system and beyond. Let's get ready for the next step. What if your data, coming from one integrated ERP system or a number of different applications, could be searched and browsed using your favorite search engine, an intuitive interface that almost everybody can use without specific information or training? What if you could access your inventory data from QuickBooks through a Web service, merge that with similar or complementary data from Oracle Financials, Great

Plains, or a proprietary application, generate XBRL GL instances using a standard technology like XSL style-sheets, and then browse them effectively with Google? Welcome to Project Nunavut.

Project Nunavut is a proof of concept that demonstrates how XBRL GL, the Global Ledger, a Web services-based approach to data distribution, and Google can work together to deliver impressively effective results in terms of interoperability, accessibility, and reusability of data. Check out Part 2 of this article next month to learn more. ■

Gianluca Garbellotto is president of Iphix, a consultancy focused on organization research and standards-based technology solutions, and a consultant to Booz Allen Hamilton. Gianluca can be contacted at gg@iphix.net.

Credit goes to Eric E. Cohen for more than just his excellent contributions to this article; to Jeff Fedor, the other half of Project Nunavut; and to Walter Hamscher for helping articulate the decisions faced by the enterprise CIO.