Collaborating with Your Customers and Partners

E-Business Integration with Progress

PROGRESS
SOFTWARE

Contents

Introduction **1**
The Keys to Enterprise Application Integration **2**
Multiple Uses for the Same Data 3
Integration Architecture **3**
Tightly-Coupled Applications 3
Loosely-Coupled Applications 4
Exchanging Data 4
An E-Business Scenario **4**
The E-Business Strategy 5
Reducing Costs by Improving Operational Efficiency 5
Opening New Markets 5
Executing E-Business Enablement **6**
Phase I: WebSpeed Gives Customers Access to Order Entry over the Internet 7
Phase II: Upgrade Enterprise Applications 7
Phase III: Integrate Enterprise Applications 9
Phase IV: Integrating External Applications 10
Conclusion **13**

Introduction

> *"... as a result of the Internet and related enabling technologies for integration and access, businesses can now more effectively leverage their IT infrastructures to interact with suppliers, customers, and employees. The resulting effects on revenue, costs, business development, and other aspects of an organization's success cannot be overstated."*
>
> – **"The eBusiness Platform: Where Application Servers and Application Integration Meet," Steve Garone, IDC Bulletin, 2001**

As companies such as eToys.com have recently demonstrated, it is considerably easier to start an e-business than it is to successfully run one. Storefronts can be put up on the Web overnight, but unless the back office applications and processes are in place to support Web sales, the effort often fails.

Existing brick and mortar companies are being forced into e-business for competitive survival. Scalable, flexible back office applications are just as important for success at existing companies as it is for start-ups. For existing companies, it is first necessary to integrate their internal applications before they can leverage the benefits of using the Internet for expanding their business.

The early history of e-business adopters is scattered with anecdotes of companies shuffling information received from an HTML form on the Web into a paper-based order-processing system, which then fed into a back-office application. These companies found their growth severely limited and many were not able to salvage enough of their business to start leveraging the Internet to solve the crisis of its internal processes.

Moving buying and selling processes to Internet has so far reaped little increased revenue—yes, businesses with a Web presence have reached new markets and opened new channels, but none that would not have been reached or opened using other approaches. The real economic benefits have been gained in the costs that have been cut as a result of streamlining business processes and integrating enterprise applications (as pointed out by Andrew Bartels, "Five Phases of E-Business So Far," Giga, May 2001). If the strategy of integrating internal applications has already accrued significant savings, consider the implications of having streamlined and automated processes not just within a single company, but across enterprises, exchanges, and collaborative communities.

What does a typical e-business strategy for a company that is on the threshold of e-business look like? Given that the Web is no longer in its infancy and has a global commercial presence, we can assume that our company already has a Web site that fulfills its rudimentary e-marketing requirements. The Web site covers the basics—it provides general corporate information, product descriptions, and a contact mechanism. In order to realize the potential the Internet offers as a technological and commercial platform, our typical company has to move beyond the electronic window-shopping phase and open its doors to customers and partners.

Moving one's business to the Web is an enormous undertaking best approached in phases. A company can begin drawing on the benefits of completed phases while moving to its goal of achieving true collaborative commerce. A successful e-business strategy includes the following phases:

- Phase One: Allow customers to complete some business transactions over the Web.
- Phase Two: Integrate critical back-office enterprise applications.
- Phase Three: Integrate legacy applications.
- Phase Four: Integrate with other enterprises.

Figure 1 illustrates how the many integration points come together in a collaborative commerce scenario.

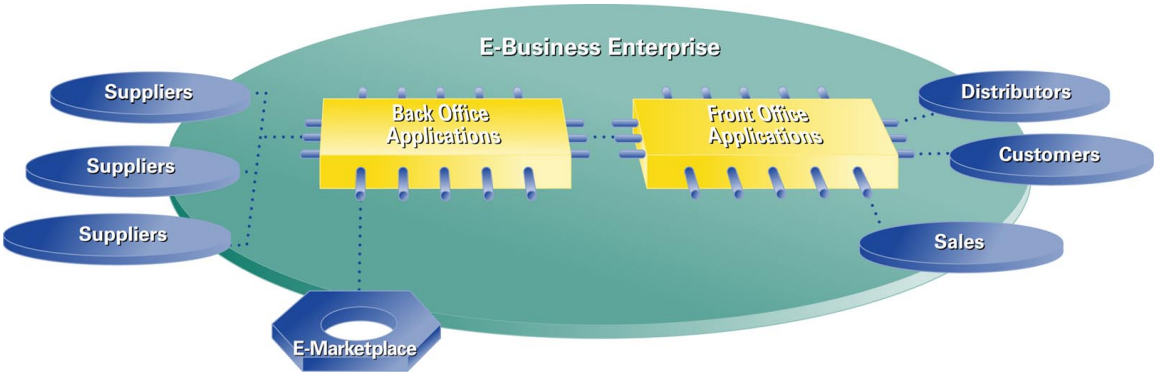

Figure-1: Collaborative Commerce and Application Integration

Your company may find itself anywhere on this continuum of e-business enablement. But at any point in the process, application integration will be a primary concern, either as you plan and design for it or as you implement and extend it. The Progress OpenEdge™ e-business platform provides the infrastructure and tools for building a solution that covers all the possibilities—integrating your own existing Progress applications, integrating third-party or legacy applications and data, integrating with partnering enterprises' applications, and creating an infrastructure that allows as yet unknown partners to open their doors to your business.

The Keys to Enterprise Application Integration

> *"Loosely coupled, asynchronous XML messaging infrastructures will dominate as the mode of communication for B2B integration"*
>
> *"By YE01, 80 percent of all Web-based technology will use XML-defined data interchange."*
>
> – **Gartner Group, 10/00**

Integration is defined as the "means of combining elements of disparate applications in order for them to interoperate." (*Steve Garone, IDC*) Two technologies are key to enabling application integration—standard format for data to ease exchange and a method for transporting information in a manner that is reliable enough to ensure the ability to perform distributed transactions—XML and Messaging. Although these technologies are becoming standard for applications deployed on the Internet platform, they are just as powerful in other network deployments.

XML is rapidly emerging as the key technology for enabling enterprise application integration (EAI) as it facilitates smoothly transferring data from one application to another. Even though applications might belong to another division or a different business and are only loosely coupled using an Internet connection, they can communicate or interoperate through XML.

XML grew out of its origins in HTML and data presentation to provide additional flexibility in defining and structuring data. It has allowed a new type of application integration to develop where applications are now loosely coupled and must accept data from almost any source. This type of integration replaces the pre-existing model of tightly-coupled integration where data exchange was precisely controlled.

Multiple Uses for the Same Data

The core functionality of XML is in line with the main driver for integrating applications, that is, realizing the benefits of efficiencies. XML allows a business to gain maximum use of its data by allowing it to be rendered into many different visualizations from the same file using XSL (Extensible Stylesheet Language). XSL allows you to describe which data fields in an XML file to display and how to display them. XML data can be transmitted to an application, a browser, a PDA and a voice reader for the visually impaired – all from the same data source. This is another competitive advantage—multiple uses for the same data. In HTML you would have to write a separate set of Web pages for each separate display of data. With XML, however, you can tune data delivery to suit internal managers, trading partners, new customers, or those just browsing a Web site. Each audience will see a different granularity based on a predetermined style sheet. This makes the data reusable and more valuable. The same Web pages can serve multiple purposes and users can get exactly the information they want, providing higher value to customers and more value out of an existing data set.

XML is but a piece of the puzzle when two companies try to communicate over the Internet. The best case occurs when businesses have collaborated and developed business processes that merge seamlessly and without intervention. In a perfect world, all business partners would share similar processes and a mutual data exchange format. By extension even future partners could be automatically identified and transacted with—all without specific prior arrangement.

Progress recognizes that XML is the emerging standard for ensuring smooth integration of applications within and between enterprises. The OpenEdge e-business platform supports XML at various levels—within the Progress 4GL, through the Internet Component Framework's SmartBusinessObjects, and by providing access to the SonicMQ E-Business Messaging Server, a leader in explicitly supporting XML message types.

Integration Architecture

> *"Messaging is also critically important to the execution and management of business processes within, and external to, an enterprise. Its relevance to e-business is extremely high and goes hand in hand with integration of business processes, legacy assets, and disparate platforms, applications, and data sources."*
>
> – **"The eBusiness Platform: Where Application Servers and Application Integration Meet," Steve Garone, IDC Bulletin, 2001**

There are two general types of integration architecture that are largely defined by the level of coordination required. These are known as tightly or loosely coupling applications. Each model has costs and benefits whether applied to integrations with internal or external applications. Both models can include XML as a data format, but the tightly-coupled model does not exploit all its advantages. One of the technologies enabling the loosely coupled architecture is messaging—in the case of Progress OpenEdge, it's the emerging Web standard, JMS-compliant messaging

Tightly-Coupled Applications

Tightly-coupled applications are applications that were designed to operate in tandem, to receive and send data in predetermined formats. Tightly-coupled applications provide the highest performance and least latency when linking two applications. However that performance comes at a price. With tightly-coupled applications there is a great deal of inflexibility, which leads to greater difficulty in programming and debugging. In addition, all coupled applications must account for any future changes to one application (or data). Lastly, there must be up-

front agreement that the design of applications match perfectly when communicating. Practically speaking this means either purchasing the same application or co-developing applications with those entities with partner enterprises. These inherent rigidities might not allow an enterprise to take advantage of the free flow of e-commerce.

Loosely-Coupled Applications

An alternative to tightly-coupling applications is to integrate applications loosely without requiring that they exchange only a pre-determined type of data formatted in a specific way. Rather, the applications are designed to accommodate XML data. Since XML carries with it the information needed to decode it, applications can make sense of data on the fly. Of course, since the exact shape of the data can't always be controlled, as the initial enforcement of business rules and validation of data takes place might take place outside of an application's context, an application must include logic for accepting data that is perhaps not complete or contains too much information. For example, you might determine that your application can accept an employee record that does not include a postal code as well as being able to handle an employee record that comes with additional insurance information.

Exchanging Data

There are several parts to an exchange of data between applications—the exchange is similar between internal applications or when exchanging data with an external application. First data from an application must be converted to XML. OpenEdge provides different ways to accomplish this—a developer can control converting data from the Progress RDBMS into XML (and vice versa) through 4GL syntax or can use the AppBuilder's XML Schema Mapping Tool in conjunction with the SmartB2BObject. Once the data is in the form of an XML document, it must be delivered to its intended application partner. OpenEdge includes an Adapter for the SonicMQ E-Business Messaging Server, a powerful, asynchronous messaging server based on the Java Message Service specification that guarantees delivery of messages. Among many features, it has built-in message queuing so that in the event of a server failure, the message will be delivered upon restoration. When an application receives a message, it must parse it (using the XML parser that Progress provides) and call the appropriate business logic.

An E-Business Scenario

> *"B2B and B2C eCommerce requires technologies to support … integration of disparate platforms; applications, legacy assets in order to leverage what works, enterprise-wide business processes for increased responsiveness and more efficient business management and administration."*
>
> – **"The eBusiness Platform: A Comprehensive Approach to Developing and Deploying Applications," Steve Garone, IDC, 2001**

Now let's look at the e-business enablement challenges and the implementation of a Progress solution for a typical company at the beginning of the twenty-first century. The RealFast Manufacturing is an established brick-and-mortar company, which produces a family of time-keeping products sold through multiple international distributors. Two years ago, RealFast determined its markets were saturated and it was facing increasing competition.

At that time, orders were taken by telephone, fax, and e-mail. The in-house sales staff entered the data in an order-entry system that ran only at headquarters. The orders were shipped from regional warehousing centers.

Each afternoon, a batch job ran against the order-entry database to produce a file of part numbers and shipping addresses for each regional warehouse.

Each warehouse ran similar, but separate, inventory and order-processing applications. These consisted of approximately 250 programs with multiple databases at each site. The host-based applications ran on a variety of platforms. The programs were written in Progress Version 7 by a staff of programmers at headquarters. There was no separation of user interface from business logic. Source code for all applications was available, but the in-house programmer skills were not sufficient to upgrade the code to an integrated e-business suite.

In to the Progress applications, headquarters runs an off-the-shelf Oracle-based General Ledger system. It accepts data from the order-entry system (item, quantity, price, terms, billing address, UPS cost) as input to produce invoices. It produces payment data as output to the order-entry system so they can close the orders when complete.

The E-Business Strategy

RealFast's e-business plan had two goals:

- Reduce costs by improving operational efficiency
- Open new markets.

The primary goal of improving operational efficiency leverages the advances made in networking and integration infrastructure and applying them to an enterprise's back office. This process of finalizing application integration within the enterprise is often a prerequisite for effectively opening applications to external customers. And in cases where it's not a clear prerequisite, completing integration reduces operating costs and allows an enterprise to have a clearer analysis of their business data, which can only strengthen the bottom line.

Reducing Costs by Improving Operational Efficiency

RealFast's plan for streamlining operations specifically calls for integrating their regional warehouse inventory systems with each other and with headquarter's order entry and General Ledger systems. RealFast foresees these integrations as having several benefits:

- Regional stock shortages and overages will be more efficiently balanced.
- Other regions can fill orders for out of stock items.
- Problem accounts can be flagged before an order is entered.

Opening New Markets

RealFast's other goal is to use the Internet to open new markets. Opening new markets is a common driver for companies moving into the e-business world. The Internet has proven to be one method for expanding o markets geographically or for reaching new pockets of customers. In RealFast's case, they plan to use the Internet to offer products directly to the end user. This will increase their potential market by adding customers in regions currently not served by existing distributors and by creating a new channel for customers who do not currently deal with existing distributors.

Executing E-Business Enablement

> *"Despite all the talk about business-to-business (B2B)integration, most integration dollars are spent on internal integration projects. Of our interviewees, 62%have purchased software tools to hook together internal apps, but only 24%have made investments in software for B2B integration."*
> – **"Demystifying B2B Integration," Simon Yates, Forrester, September 2000**

RealFast has developed a technical strategy for executing their e-business plan. Their goal is to establish an e-business presence as quickly as possible, then to transform their enterprise application infrastructure so that it is a strong foundation up which to continue to expand their e-business opportunities. After providing their customers with access to the applications that impact them the most, RealFast wants to position themselves technologically so that they can form e-business partnerships with suppliers and distributors.

RealFast analyzed their current enterprise application suite in terms of e-business potential and possible weaknesses. Their response was a phased approach that combined implementing technology to gain immediate benefits while pursuing the long-range goal of designing an e-business suite of integrated enterprise applications that could in turn integrate into their suppliers' and distributors' applications.

Phase I. The Order Entry application was chosen as the application which would have the greatest impact on tightening relationships with current customers and gaining new ones. Through it, RealFast can start expanding their direct sales, develop regional markets, and provide a higher level of customer service. To achieve the aim of "speed to market," they have decided to create an add-on module running WebSpeed that would give customers access over the Web to the Order Entry application logic and data. The more time-consuming task of Web-enabling the Order Entry application itself was planned for Phase II.

Phase II. Web-enable the Order Entry application along with the back-office applications, making sure that all enterprise applications are integrated into one cohesive suite, including an Oracle-based legacy application. Internet technology and e-business infrastructure made this integration an imperative as well as made it a possibility. OpenEdge provides the blend of WebClient and browser-based clients as the building blocks of their e-business infrastructure.

Phase III. Integrate an Oracle-based legacy application. The Progress DataServer for Oracle makes the application database's integration appear seamless.

Phase IV. As XML is the emerging standard for exchanging data among e-business applications, RealFast will add to their enterprise applications XML messaging services using SonicMQ and the Progress SonicMQ Adapter so that they can offer the external applications of their suppliers and distributors an integration point.

By applying the Progress Version 9 Universal Application Architecture and core components of the OpenEdge e-business platform, such as WebSpeed, Progress AppServer, and WebClient, RealFast was able to execute Phase I and Phase II last year. They are currently in the process of finalizing the integration of their Oracle-based system and implementing a messaging service to facilitate integration with partners' applications (Phase III and Phase IV).

Phase I: WebSpeed Gives Customers Access to Order Entry over the Internet

The first phase of RealFast's e-business integration was to add a WebSpeed module that allowed external users to access the Order Entry application through the Internet. These were the steps involved in implementing the WebSpeed module:

1. Using WebSpeed, develop a browser-based shopping cart interface to the Order Entry application that allows customers to enter orders and query the status of their orders. The two activities that this module supports are not transaction-intensive, therefore they will not require many requests going back and forth over the Internet. These activities lend themselves well to an HTML-based interface typical of today's Web applications.
2. Design the HTML interface displayed by the user's Web browser. Developers can use their favorite HTML authoring tool or they can write their HTML and JavaScript in WebSpeed Workshop.
3. Use embedded SpeedScript (the scripting language based on the Progress 4GL) or Java Script to add data-entry validation logic to the HTML interface.
4. Write the data-manipulation logic for writing records and performing simple queries to the Order Entry database. This application logic is executed by the WebSpeed Transaction Server.
5. Build in support for region-specific interfaces. Create region-specific HTML interfaces. For localizing data display, rely on Progress' support for internationalization. WebSpeed and the Progress database support all the world's business languages for a native look and feel to error messages and data formats.
6. Implement full security to protect customer information and the enterprise. WebSpeed's architecture gives RealFast the freedom to integrate its choice of security solutions, including firewall, authentication, authorization, and encryption technologies. This security for this module involves a combination of SSL, SecureID, and application-based customer identification. The customer is identified at login so that the appropriate business terms can be displayed and used.

Phase II: Upgrade Enterprise Applications

RealFast's critical enterprise applications targeted for Phase II of the e-business integration plan are its order-entry application and its regional inventory management systems. Before the order-entry application can be integrated, it has to be re-cast in the UAA model. Implementing this phase had the following steps:

1. In the order-entry application, separate business logic (BL) from their user interface (UI). This is a necessary step because the separation makes it possible to use n-tier architecture. You run your business logic on the AppServer, leaving the UI logic to be rendered by whatever client technology you choose. Separating BL from UI also simplifies changing or adding to the UI since the business logic is not affected. Developers can build a 4GL character client, Windows GUI client, WebClient GUI, HTML for a Web browser or PDA, and Java or ActiveX client interfaces, depending on customer needs.

   An important step in separating BL from UI is finding specific pieces of the existing code that can be replaced by Version 9 components. Specifically, that means finding points of data access, such as FOR EACH and FIND statements, and replacing them with SmartDataObjects (SDO) and dynamic queries. The benefits of using dynamic objects are that they:

   - Are reusable in multiple situations
   - Reduce programming by reusing a single object for multiple purposes
   - Create a more flexible and maintainable application

- Simplify the enforcement of standards
- Reduce client r-code size
- Allow for run-time customization so that you can translate labels, titles, etc., hide or disable fields based on security, build menus of available components, and modify an application's look and feel depending on user profiles or changing company standards

2. Use the Progress AppServer's n-tier architecture to distribute the application and make maximum use of existing hardware infrastructure.
   RealFast's reason to deploy on the Internet is to gain more customers, which will increase usage. The n-tier architecture scales to accommodate both short-term and long-term growth. Distributing an application onto an n-tier architecture has a major impact on ease of deployment (installation and subsequent maintenance). The WebClient's IntelliStream™ technology simplifies upgrades with a server-based provisioning system that enables users to download only those application components that have changed on an as-needed basis. It is not necessary to distribute a completely new application because you have changed part of the business logic.

3. Use WebClient across the Internet to access the AppServer and the inventory management system from the regional locations. The benefits to this strategy are that there is no new technology to be learned to support the Web-enabled enterprise applications. Both the front-end UI and the back-end Business Logic written in 4GL. This makes it possible to salvage portions of the existing applications. The WebClient combines a thin-client profile with the ability to provide rich GUI functionality, such as the full range of GUI objects, OCXs, OLE automation, a SmartObject interface to generate and consume XML documents. In addition, the WebClient makes calls to the asynchronous AppServer, allows for batch processing, fully implements Progress internationalization features, and supports access to non-Progress data sources through Progress DataServers.
   A WebClient implementation minimizes the amount of data and frequency of data exchange between the client and the server. This is a very important requirement for high-performance and scalable applications. It is critical when using a WAN or public communications infrastructure (the Internet).

4. Secure the enterprise by implementing security. WebClient takes advantage of the Internet's Standard Infrastructure (HTTP, SSL) to provide security, including over-the-wire encryption with SSL support, support for digital signatures/certificates with Microsoft's Authenticode Framework, and cache for authentication information (single sign-on).

5. Consolidate databases and restructure them by using storage areas. A single database, or reduced number of databases, is easier to maintain, change, backup, and replicate. The consolidation of disparate databases may include redundant information that must be integrated using conversion routines. The conversion routines must account for different data types and field sizes used to represent the same data in different databases. Dry runs of the process are required before the real merge in order to work out bugs and unintended side effects. Full backups of the original databases are also required prior to beginning the real merge. Once the databases have been consolidated, data-maintenance routines can be built in the Progress AppBuilder.

6. Test existing batch routines, reports, and validation utilities to make sure they still work. There are no required modifications.

Phase III: Integrate Enterprise Applications

This phase calls for integrating the inventory management system and the Oracle-based General Ledger system using the Progress DataServer for Oracle. Progress DataServers provide 4GL developers with the ability to use the Progress component-based development environment with other database engines such as Oracle. Programmers can take advantage of the Progress DataServer for Oracle and their skills in the Progress Development Environment, to write procedures to read, write and update data shared between the Progress Database and Oracle. This approach allows RealFast's 4GL developers to integrate data stored in the different data managers by the applications, correctly maintain order status and enable correct billing. By providing consistent application behavior, such as locking, scrolling, sort order, and case sensitivity, DataServers simplify application development and deployment enabling the application developer to write applications against a range of data sources. The Progress Data Dictionary hides much of the complexity normally associated with developing and maintaining database definitions, application defaults, and business rules.

RealFast want to link these two applications such that when a customer enters an order in the order-entry application, that activity triggers the General Ledger system so that a customers' credit and billing information is automatically calculated and recorded. It also updates the invoice information accessible through the order-entry application. Since all developers involved in the integration work for the same company and can easily share the design of the applications and the structure of the databases, they have decided to tightly couple their applications. This will give them the least latency and the highest performance for their customers.

The architecture they have decided on involves connections at two points (see Figure 2):

1. Order entry 4GL API to General Ledger API. This is an A2A communication link that can use HLC (Host Language Calls), OLE automation, Remote Procedure Calls (RPC) et.
2. Order-entry 4GL application to the Progress DataServer for Oracle to the General Ledger Oracle database. This is an application-to-data source connection using a direct data link.

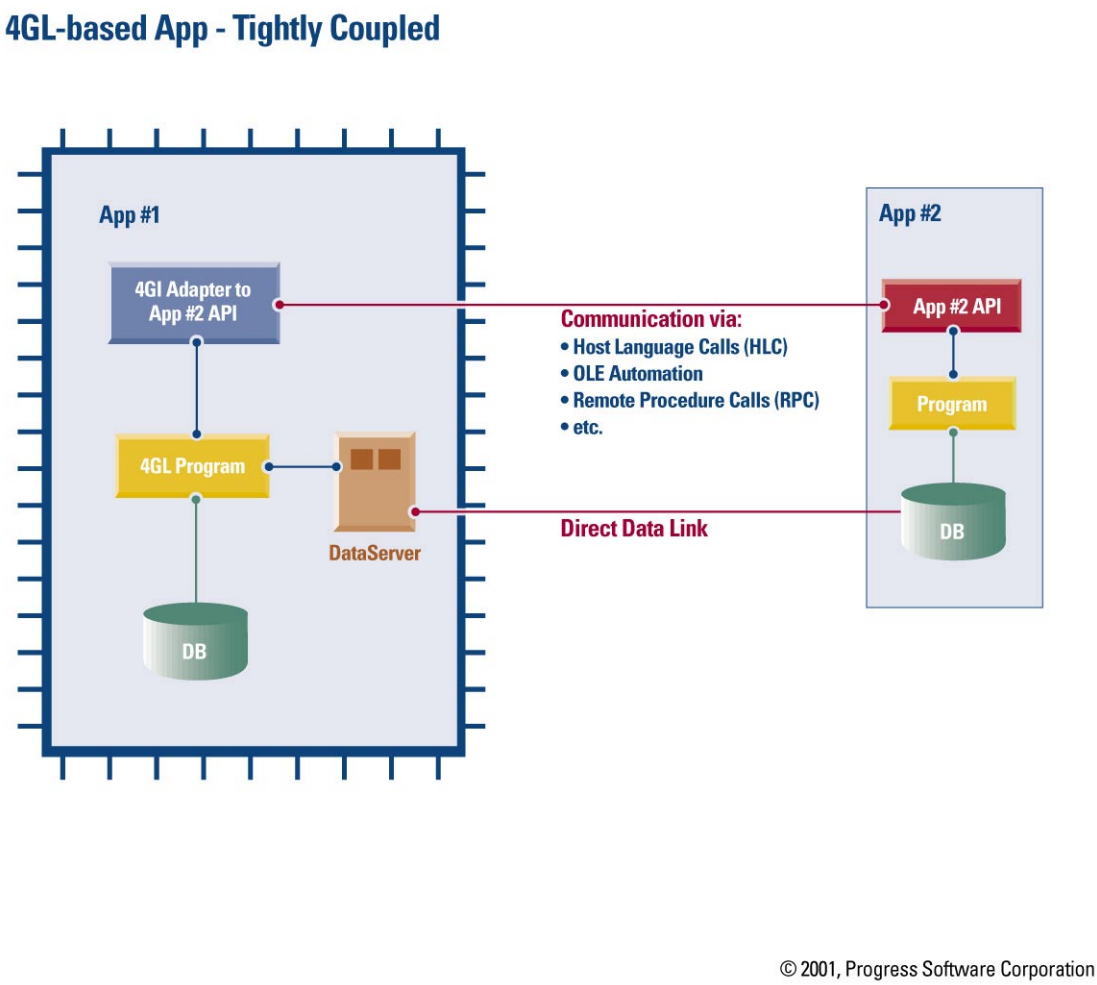

Figure-2: UAA-Based Application Tightly Coupled with Internal Non-Progress Applications

Now when John places an order into the order-entry application information is funneled through the 4GL and HLC to the API of the warehouse application. The General Ledger acts immediately on the request, determining appropriate discounts and calculating final account information. An update is sent back through the same channel where the information can be available to customers in minutes. An add-on module running in WebSpeed automatically posts any new orders or any change in order and account status to a Web site accessible by inside and outside parties with just a browser.

Customers can receive immediate account information. Sales and customer service representatives have an up-to-date profile of customers' account statuses. Through business-intelligence applications, RealFast executives can now have real-time access to that same information.

Phase IV: Integrating External Applications

The final phase of RealFast's e-business plan is to make it possible to integrate with other enterprises' applications. Just as in many real-world cases, this phase is planned in the abstract, but its implementation is jump-started by direct requests from partners. The largest distributor of RealFast Manufacturing products would like faster response to requests for quotes and inventory status.

RealFast has a variety of choices for accessing their distributor's applications or data. The distributor's application happens to be written in Progress, so it can be directly linked to the Progress order-entry application at RealFast very easily. The application can also be tightly-coupled to the General Ledger system much the same way the order-entry system is by using a VPN (virtual private network). However due to the uncertainty of the reliability of the Internet provider and the added complexity of direct coupling, John decides on a loosely-coupled arrangement. Moreover, by choosing a loose coupling, RealFast can more easily accommodate future integration requests, many of which will involve non-Progress applications. There are also long-term efficiencies to consider as tightly coupling applications implies that updating either application may require changes to the other to maintain synchronization—a complex, if not impossible, undertaking when dealing with other companies or trade exchanges.

Using SonicMQ, the Progress Adapter for SonicMQ, and the SmartBusinessObjects that Progress provides makes it easy to send XML messages to the RealFast order-entry system. Guaranteed delivery by SonicMQ takes the worry out of relying on an Internet connection—in case of a connection failure, no messages are lost. Figure illustrates how these applications interact.

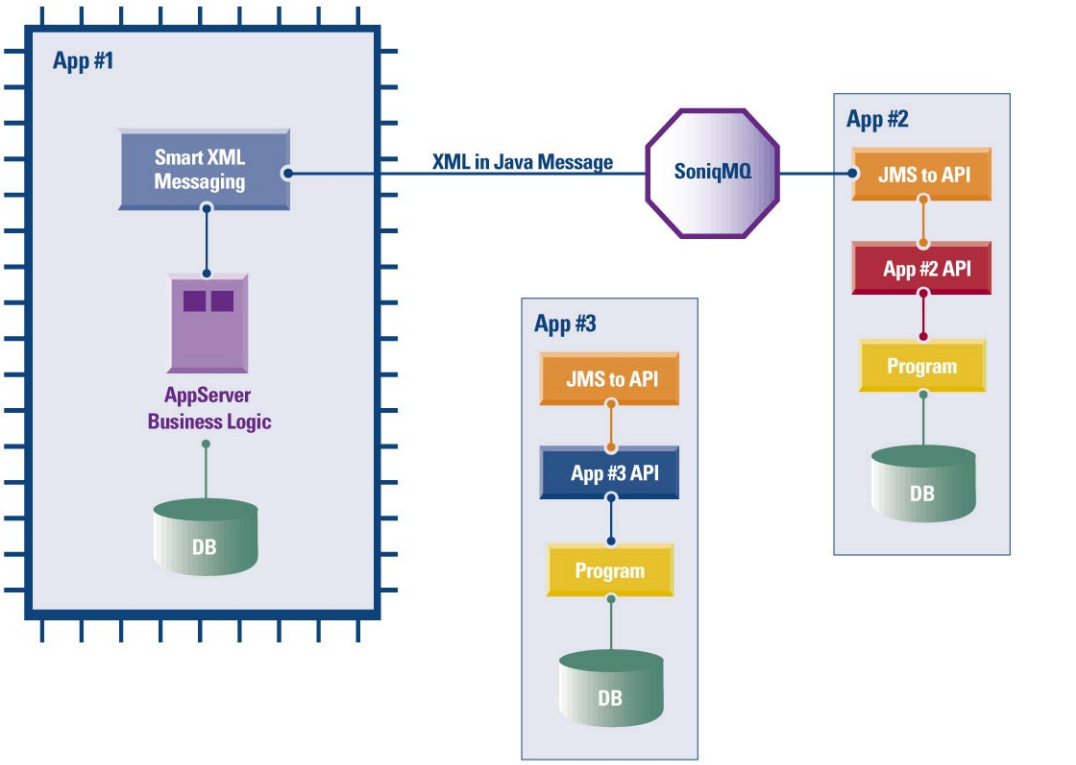

Figure-3: UAA-Based Application Loosely Coupled with External Applications Through SonicMQ

Used together, the SmartBusinessObjects handle the transformation of Progress data to XML, the wrapping of XML into messages and the transfer of that data to SonicMQ for delivery as a Java Message Service. JMS is specifically designed to be a highly-reliable, asynchronous message delivery service for the Internet. Like XML, JMS is rapidly becoming ubiquitous on the Web. Figure 4 illustrates how the SmartBusinessObjects work together to implement E-Business Messaging with SonicMQ.

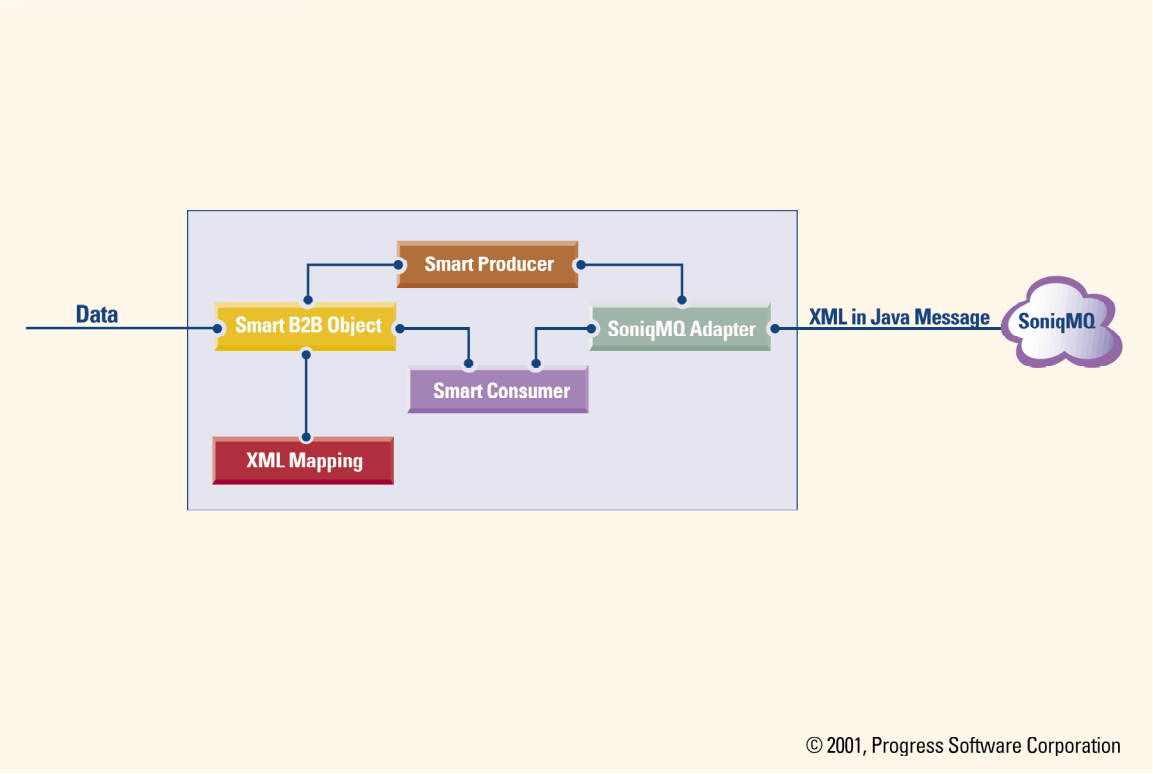

Figure-4: Progress Smart XML Messaging, 4GL-to-JMS API, and 4GL-to-XML Mapping

It should be noted that developers can complete XML integration in Progress can be without using the SmartBusinessObjects presented here. For further information on the 4GL functions that control working with XML data and JMS messaging, please see the Progress documentation, *Progress on the Web*, *Progress External Program Interfaces*, and the *Progress Language Reference*.

Table 2 summarizes the points of application-to-application access demonstrated by this scenario. The specific Progress products involved in the scenario are in bold.

Table 2: Summary of Enterprise Application Integration

Activity	Integration Scenario	Technology	Application
RealFast customer places an order in the order-entry system, which then updates the General Ledger System.	Internal	4GL Client, AppServer, Progress DataServer for Oracle, HLC, Non-Progress application, Non-Progress Database	Order Entry to General Ledger
RealFast customer makes an account query from the order-entry application directly to the General Ledger Database.	Internal	4GL Client, AppServer, Progress DataServer for Oracle, Non-Progress Database	Order Entry to General Ledger

Activity	Integration Scenario	Technology	Application
WebSpeed queries the order-entry Progress database and posts any order change to a known Web site.	Internal	WebSpeed, Progress RDBMS, Web server, Web browser	Internal batch to external Web site
WeGotIt places an order from their own application and Web site.	External	4GL client, AppServer, XML, SmartObjects or SonicMQ Adapter, SonicMQ messaging server, to RealFast Application	Distributor's Order Entry

In addition, with full support for the ANSI SQL-92 Entry Level specification, Progress RDBMS products integrate with enterprise applications, tools, and numerous third-party data management systems. Featuring a state-of-the-art, cost-based query optimizer, the SQL-92 language processor incorporates APIs for ODBC, JBDC, and Embedded SQL/C, and provides SQL extensions for robust solutions using the SQL-92, ODBC, and JDBC industry standards.

Access to the Progress database from third-party reporting products (Actuate), data mining and business intelligence tools (CorVu), other languages (like VB, C++ and Java), desktop utilities (Microsoft Office), and in general, any JDBC- or ODBC-compliant product is provided by the Progress ODBC and JDBC drivers.

Conclusion

There is technology currently available that makes it quite simple to bring your existing business application infrastructure to the Web, at least a part of it. When you choose to open part your business processes to customers over the Web, for example, by giving them access to a catalog and an order-entry application, by exposing that corner of your business processes to the Web, you are, in effect, potentially exposing much more of your infrastructure. By quickly implementing a the first phase of an e-business strategy, you will get to the Web, but you won't get very far in terms of true commercial success. What happens when you receive more orders than you can process or fulfill? When you gain more customers than you can serve with any level of quality? When you reach more global markets, with their specific currency and tariff structures, than your financial systems can accommodate? When new opportunities for partnerships come with the requirement that your applications interoperate with theirs?

Your business will only be as capable of acting on the Internet's opportunities as your enterprise application infrastructure is resilient to their demands. One way to achieve that resiliency and flexibility is by integrating enterprise applications so that all aspects of your operation are streamlined and synchronized, allowing your enterprise to act as one unified entity. Using standards-based technology, such as XML and JMS messaging as provided by OpenEdge, to build that infrastructure allows you to extend the benefits of integration to all your business transactions, including those with your partners in collaborative commerce.

Corporate Headquarters
Progress Software Corporation, 14 Oak Park, Bedford, MA 01730 USA Tel: 781 280 4000 Fax: 781 280 4095

Europe/Middle East/Africa Headquarters
Progress Software Europe B.V. Schorpioenstraat 67 3067 GG Rotterdam, The Netherlands Tel: 31 10 286 5700 Fax: 31 10 286 5777

Latin American Headquarters
Progress Software Corporation, 2255 Glades Road, One Boca Place, Suite 300 E, Boca Raton, FL 33431 USA Tel: 561 998 2244 Fax: 561 998 1573

Asia/Pacific Headquarters
Progress Software Pty. Ltd., 1911 Malvern Road, Malvern East, 3145, Australia Tel: 61 39 885 0544 Fax: 61 39 885 9473

PROGRESS
SOFTWARE

www.progress.com