TDWI

MONOGRAPH SERIES

MAY 2006

Embedded Analytics: Closing the Loop Between Operational and Analytical Applications

By Wayne W. Eckerson

Director of Research and Services
The Data Warehousing Institute

SPONSORED BY

IBM®

tdwi
THE DATA WAREHOUSING INSTITUTE

About the Author

WAYNE W. ECKERSON is the Director of Research and Services for The Data Warehousing Institute (TDWI), a worldwide association of business intelligence and data warehousing professionals that provides education, training, membership, certification, and research. Eckerson has 17 years of industry experience, and has covered data warehousing and business intelligence since 1995.

Eckerson is the author of many in-depth reports, a columnist for several business and technology magazines, and a noted speaker and consultant. His book, *Performance Dashboards: Measuring, Monitoring, and Managing Your Business,* was published by John Wiley & Sons in October 2005. He can be reached at weckerson@tdwi.org.

About TDWI

The Data Warehousing Institute (TDWI), a division of 1105 Media, Inc., provides in-depth, high-quality education, training, research, and certification for business intelligence (BI) and data warehousing (DW) professionals worldwide. TDWI is dedicated to educating business and information technology professionals about the strategies, techniques, and tools required to successfully design, build, and maintain data warehouses and BI solutions.

The Current State of Business Intelligence

Business intelligence (BI) consists of tools, technologies, and processes that deliver information and insights to business users to improve decision making and optimize performance. BI has grown in strategic importance in the past decade, as executives recognize that providing critical business information in a timely fashion to employees, customers, and suppliers is a key to success.

As a result, BI is now the highest ranked technology priority among CIOs, according to a 2006 survey of 1,400 CIOs by the research firm Gartner Group. The survey also showed that CIOs plan to increase their BI budgets by an average of 4.8 percent in 2006. The Gartner Group estimates that the market for BI products is currently $2.5 billion and will grow to $3 billion in 2009.[1]

The potential of BI so far exceeds its achievements.

CHALLENGES. Despite its impressive market size, BI has not yet reached its full potential within most organizations. Many are struggling to deliver the benefits of BI to more than just a handful of users who have the technical and business literacy to exploit the current generation of BI technology. In addition, most BI solutions only let users analyze the business through a rear-view mirror: The solutions provide historical data about what happened in the past, not real-time data about what is happening right now. Real-time information gives executives greater visibility into business operations and enables workers to take immediate action in response to problems and issues.

BI Tools Must Be Easier to Use. For BI to penetrate deeper into organizations and deliver strategic benefits, BI tools must be easier to use and provide insights into business events as they happen. Of course, this is no news to BI vendors, who have made sizable strides in recent years in simplifying their BI tools for use by "everyman"—from executives and managers to front-line staff, customers, and suppliers. BI vendors now offer role-based views, dashboards and scorecards, and more recently, Google-like search tools. Yet, more needs to be done.

BI Must Deliver Timely, Contextual Operational Data. On the data side, BI professionals are embracing the notion of "operational BI." They are modifying classic batch-oriented data warehouses to support event-driven processing so they can deliver real-time information to users within historical context. They also use federated query capabilities, Web services, and service-oriented architectures to cull

[1] "Business Intelligence Software Market to Reach $3 Billion in 2009," *CRM Today*, February 8, 2006.

operational data in real time from disparate systems. While these approaches help, they are not enough to fully operationalize BI and deliver insights on demand.

The Promise of Embedded Analytics

In the end, the best way to simplify and operationalize BI is to embed it directly into operational applications and processes that drive the business. This is the definition of embedded or "inline" analytics.

Nothing New. Embedded analytics are nothing new. Organizations have embedded BI functionality into applications and business processes for years. For example, developers often create reporting capabilities from scratch when building custom applications. Portals display charts and tables generated by reporting tools inside portal windows or "portlets." Microsoft Office applications maintain live links to reports stored on BI servers. Extranet applications integrate with BI engines that let online customers view and analyze personal accounts and activity. Online applications embed predictive models that score customer transactions in real time to detect fraud, cross-sell products, evaluate risk, or assess credit worthiness.

Most embedded analytics to date have barely scratched the surface of what is possible.

However, most embedded analytics to date have barely scratched the surface of what is possible. Usually, the analytics are fairly primitive, displaying canned views of existing reports with little ability to drill down, publish views in other formats, or compare other data. The more compelling analytically driven applications are implemented by leading-edge companies with deep pockets and legions of skilled developers who code, debug, and test monolithic applications that are time-consuming to build and costly to modify.

The Future of BI. Today, however, visionary vendors and BI professionals are conjuring new ways to blur the lines between analytical and operational applications. They trumpet the benefits of composite applications, process-driven BI, business activity monitoring, BI services, operational dashboards, integrated business application suites, software-as-a-service, and open source BI, among other things. With new development techniques that make embedding analytics into business processes and applications as easy as dragging and dropping objects onto a workbench, the future course of BI could change radically.

Users may no longer realize they're using BI tools.

Rather than using stand-alone BI toolsets that require setup and training, business users will leverage embedded BI functionality that is an integral part of a larger application or package. Users will no longer shift software contexts when moving from operational processes to analytical ones. Currently, they must exit an operational application—and their train of thought and productivity—pull up a BI tool, find the appropriate report, analyze it, and then reenter the operational application and make appropriate updates based on the insights they gleaned in the BI tool.

In the future, however, users will gain analytic insight within the context and flow of a single, process-driven application. At this point, BI simply slips into the background

of a primary application that users use to do their jobs. Users may no longer realize that they're using distinct BI tools to access and analyze information.

BI as a Service. To switch on a lamp, you must first plug it into an outlet that taps into the electrical grid. Like electricity, BI is destined to become an enterprise service that users and applications tap into to deliver information and insights to users on demand. Embedded analytics transforms BI from sets of stand-alone products to enterprise services that make BI easier to use and pervasive.

BI tools can launch operational tasks—a kind of reverse embedding.

BI as a Container. Conversely, applications built using BI tools will serve as vehicles to launch operational processes and tasks—a kind of reverse embedding. For example, many companies now use dashboards to monitor key business events, trigger alerts and workflows, and execute tasks within operational applications. In addition, some packaged software vendors now use a dashboard as the central metaphor for delivering the role-based views of tasks and information required to manage processes in and across multiple business departments.

Benefits of Embedded Analytics

Embedding BI into core applications and business processes has many benefits:

1. **Makes BI more pervasive.** By embedding analytics into core operational applications and processes, organizations deliver the benefits of BI to many more employees without having to purchase BI tools for each user.
2. **Makes BI easier to use.** Embedded analytics are easier to use because they conform to the "look and feel" of operational applications that users use on a daily basis. In addition, functionality is tailored to individual roles. Embedded analytics usually require little training.
3. **Makes BI actionable.** By linking BI to operational processes and data, users gain a better understanding of the data and its business context, enabling them to not only spot and fix problems before they escalate out of control, but also identify and seize new opportunities before they vanish.
4. **Makes BI more flexible.** BI services can be incorporated into new and existing applications more easily, avoiding bottlenecks that plague report development in many companies and cause users to develop shadow IT systems.
5. **Optimizes business processes.** BI can monitor the status of key business processes, alert individuals when performance exceeds predefined thresholds, and trigger alerts, workflows, and tasks to streamline processes and optimize performance.
6. **Closes the loop.** Embedded BI integrates operational and analytical functions so information drives action in a virtuous cycle that helps companies better understand what drives their business and adapt more quickly to new and changing conditions.

Dimensions of Embedded Analytics

Defining Embedded Analytics

As mentioned, embedded analytics build BI functionality into the fabric of operational applications and processes that drive the business on a daily basis. But we need to further unfold this definition to get clarity on the scope and dimensions of embedded analytics.

Operational Applications and Processes. By *operational applications,* we mean software that enables users to execute and manage business processes in various functional areas of an organization, such as sales, service, and marketing. Collectively, these applications may support a cross-functional business process, such as a customer relationship management initiative, that taps into sales, service, and marketing functions within one or more departments.

One or more steps in either the departmental-specific or cross-functional processes may involve accessing BI services to deliver information required by an individual or an application to take action or make a decision. This information-gathering step requires integrating BI within custom-designed solutions or commercially available business process management (BPM) environments.

The diversity of embedded BI functionality depends on business requirements and the richness of APIs.

Analytics. *Analytics* refers to a comprehensive set of BI functionality currently available through stand-alone BI packages. This includes ad hoc querying, reporting, dimensional analysis (slicing/dicing or pivoting), "what-if" modeling, monitoring and alerting, visualization, forecasting, and predictive analytics, among others. The level and type of BI functionality embedded into operational applications depends on business requirements and the richness of the application programming interfaces (APIs) used to embed BI functionality. Not all applications require organizations to embed a complete set of BI functionality, but more functionality is generally better than less.

Embedding. Operational applications dynamically call BI functions or services, which may reside locally within the same application or remotely on a different server. Local calls are handled through some sort of interprocess communications facility (a DLL or internal API) while remote calls are handled by an API, which may be proprietary (a BI tool API) or standard (XML/SOAP). The operational applications support one or more business processes that may be automated, semi-automated, or entirely manual. In semi-automated and manual processes, an individual rather than an application executes the call to the BI server or service (submits a query or requests a report).

Reverse Embedding. In turn, BI applications or services may make calls to operational applications to execute various tasks, using SQL, stored procedures, e-mail, or URL links. BI engines often embed business logic that steps users through the process of analyzing exception conditions and taking the appropriate action. These BI engines often invoke processes in other applications, such as updating a database, submitting an e-mail message, or launching a stored procedure. As

mentioned earlier, many companies use dashboards and scorecards to monitor business activity and automatically trigger alerts and actions to proactively deal with issues as they arise.

Bundling BI tools into application packages is not embedded analytics.

Pseudo Embedding. Some people use the term *embedded analytics* or *embedded reports* to refer to BI tools that are bundled within other applications. Bundled products typically need to be installed separately, and some come with a restricted use license. For example, many application vendors provide BI tools, such as Crystal Reports, with their packages. Many BI vendors also bundle separate and distinct BI tools within the same package, and then link back and forth between the tools using hyperlinks, portal windows, or by popping up separate windows. This is a poor man's integration and differs from embedded analytics. However, a software vendor that negotiates an OEM (original equipment manufacturer) license for a BI product and integrates it seamlessly into its core application with a single install and common object model has, in fact, embedded the secondary product.

Approaches to Embedded Analytics

Given the scope of our definition of embedded analytics, there are several ways to embed BI into applications. We have already hinted at most of these.

1. Local Services. Developers can embed BI services (reporting, OLAP, query engines) and libraries of BI objects and components into a single application. Here, the code for both the operational application and the BI services reside within the same codeset and run on the same server and platform. Developers embed queries to display reports to users or execute an information-gathering step within an existing process governed by the application.

Open source BI tools aid programmers who embed analytics from scratch.

Traditionally, most Java and .NET developers are not familiar with BI tools and concepts. Consequently, they often create BI capabilities from scratch using their programming language and development tools of choice to augment their applications. The availability of open source BI tools from the Eclipse Foundation (Eclipse BI Reporting and Tools), JasperSoft, and Pentaho now make it easier for developers to embed BI source code into their applications. (See Figure 1.)

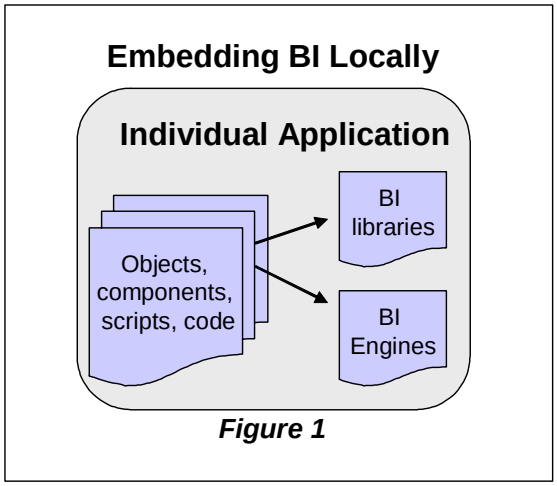

Figure 1

2. Remote Services. The other option is for developers to create applications that call BI services residing on remote servers via proprietary or standard APIs over a network connection. Here, the application code is separate and distinct from the BI code, and both reside on different servers and platforms. (See Figure 2.)

For example, portals call remote BI services to render reports or charts within a portal window or "portlet." BI vendors or in-house developers develop BI portlets using proprietary or standard portal APIs, such as JSR 168, which emanated from the Java Community Process. Another example of remote services is Microsoft Smart Clients, which enable MS Office applications such as Excel and PowerPoint to maintain live connections with reports residing on remote BI servers.

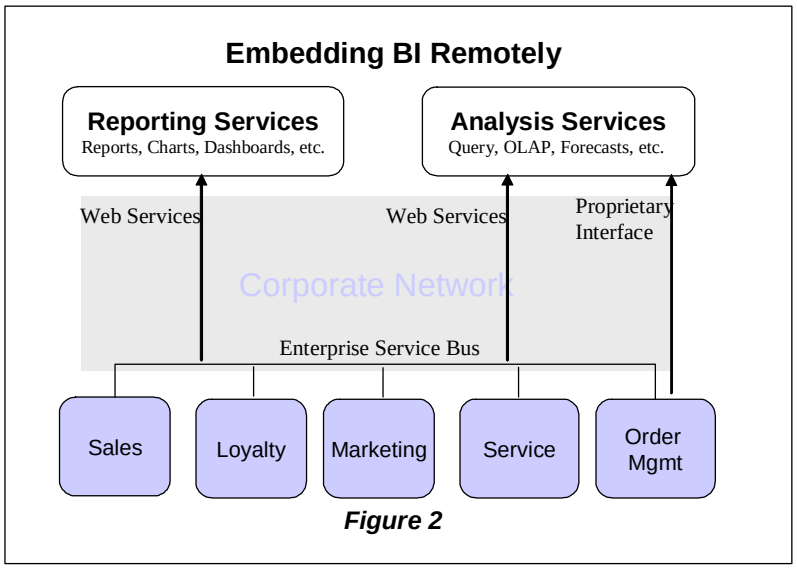

Figure 2

Composite applications blend analytical and operational components in a transparent fashion using service-oriented middleware.

In addition, leading packaged software vendors such as SAP and Oracle promote the notion of composite applications that enable developers to transparently knit together applications, services, and components providers using application middleware services. These include SAP's NetWeaver and Oracle's Fusion, which make heavy use of Web services standards.

Business process management (BPM) engines can also call remote BI services to execute an information-gathering step within a business process. For example, an online loan application may call a third-party service to obtain a credit report about a customer and deliver it to a loan officer via e-mail. And of course, there are the Web services themselves, which enable applications to call remote BI services using standard interfaces and network protocols.

3. Modular Embedding. As mentioned earlier, software vendors can embed BI products into applications or suites through custom integration within an OEM relationship or by creating the BI capabilities from scratch. (See Figure 3.) Many applications already use OEM BI components to deliver full-featured solutions. BI vendors themselves use OEM charting, visualization, annotation, and other components that are not their core competency but that other vendors are willing to provide.

New integrated business applications for the small and midsize business market, such as NetSuite or Microsoft Dynamics application suites, offer integrated BI capabilities, usually in the form of dashboards that drive core business processes. (Here, analytics are the brain of the application that coordinates business activity within and across business functions and departments.) Also, emerging software-as-a-service offerings, such as those from Salesforce.com, now incorporate BI functionality as embedded services that look and feel like they are part of the original application.

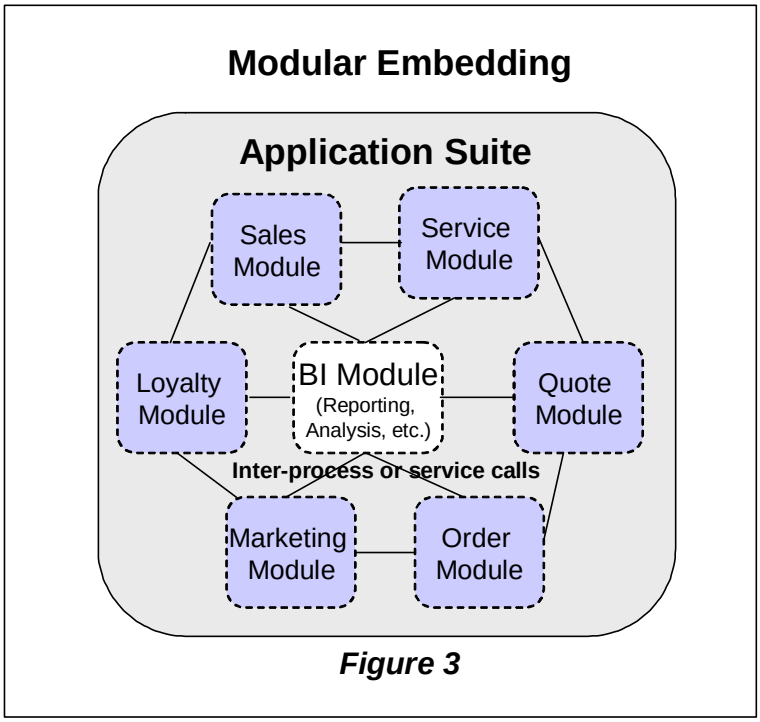

Figure 3

4. Event Processing. As discussed earlier, BI services can launch processes and execute tasks within operational systems in response to business events. For example, many BI dashboards can monitor business events either in real time using event-driven processing models, or in batch through periodic polling of upstream systems. In both cases, the dashboards track business performance against pre-defined goals in the form of key performance indicators (KPIs). The dashboards then trigger alerts that notify business users of exception conditions or automatically initiate workflows or execute operational tasks to address a situation. Some vendors now integrate BI and BPM software to automate business processes in this way. (See Figure 4.)

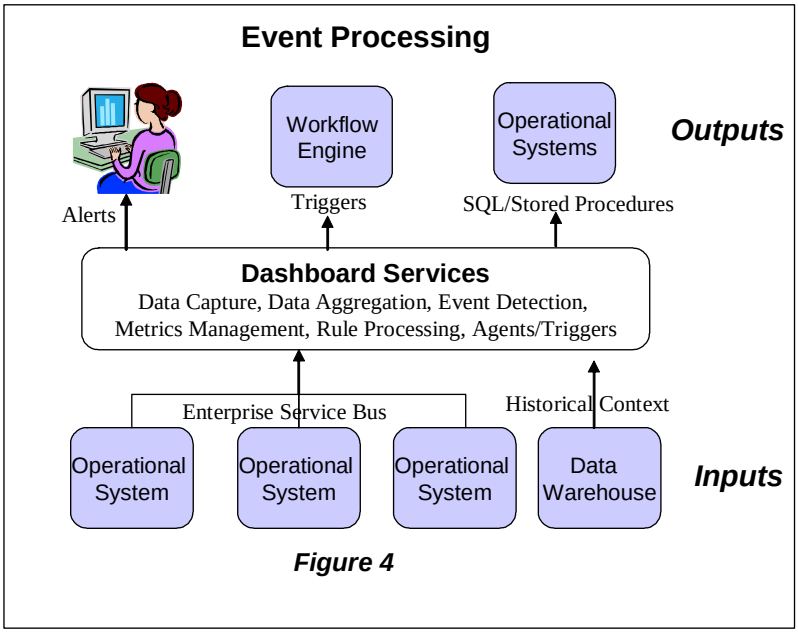

Figure 4

Tactical dashboards monitor daily and weekly events and trigger manual processes.

Tactical dashboards. Tactical dashboards, which are updated daily or weekly, trigger alerts to exception conditions and then guide users on the appropriate actions to take.

For example, a grocery manager uses a tactical dashboard to track his daily progress toward achieving corporate objectives for the store, such as increasing the percentage of staff who greet customers in the aisles. If the manager notices a yellow or red light next to this KPI, he can drill down to view staff members who failed to greet the secret shoppers who compiled the data. The dashboard then provides a list of steps the manager should take to remedy the problem, including talking to the staff person or asking him or her to watch a training video or attend a Web-based training course. (See Figure 5 to view the grocery store manager's dashboard.)

Tactical Dashboard in Action

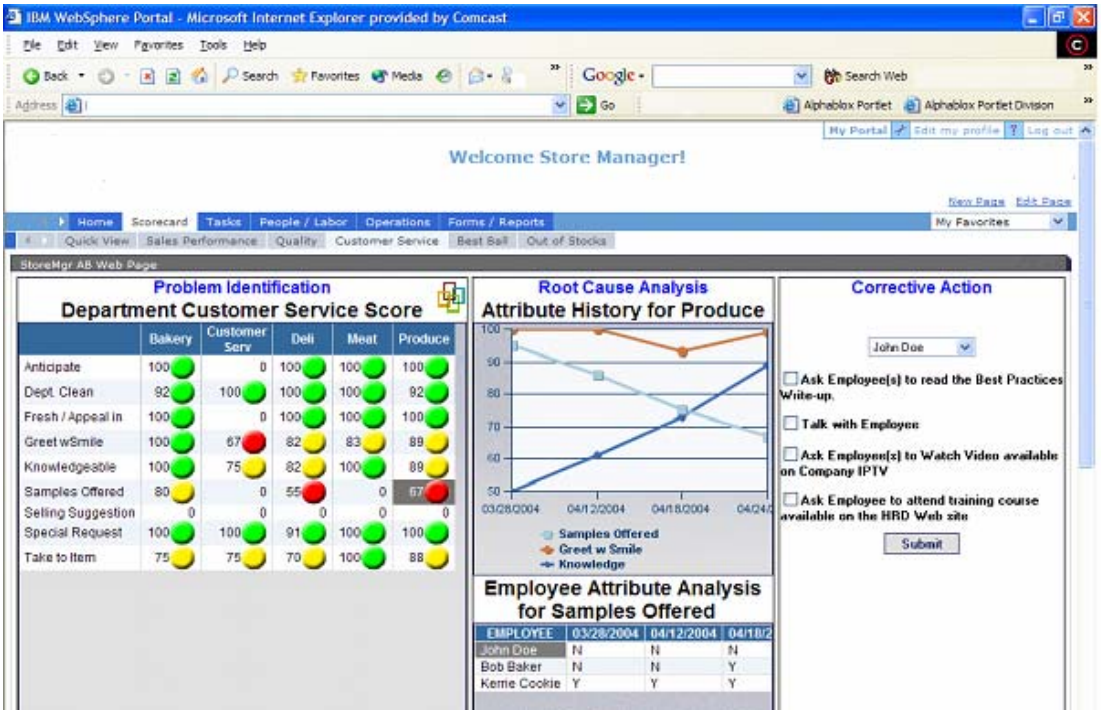

Figure 5

Operational dashboards monitor events as they happen and trigger automated responses.

Operational dashboards. Operational dashboards, on the other hand, monitor events as they happen. They do this by capturing events in real time, either from an organization's messaging backbone (an enterprise service bus), via data replication, or by continuous polling (querying.) The dashboard engine then aggregates and correlates the events in a dimensional schema, compares them to historical data in a data warehouse, a local cache, or some other repository, and calculates the relevance of events based on predefined metrics of acceptable performance. When performance exceeds thresholds, the dashboard automatically triggers alerts, launches workflows ("Send an e-mail message and report to the head of eastern region sales and request an update in three days") or executes tasks ("Submit a purchase order to supplier X for 50 widgets").

These automated actions are governed by a rules engine that business users define within the dashboard or an external BPM system. Some industry experts refer to this type of event-driven analytical system that monitors key business processes as business activity monitoring (BAM). This type of reverse embedding is critical for helping companies streamline business processes and optimize performance. Thus, it's a critical piece of most business process management strategies.

Methods for Developing Embedded Analytic Applications

It should be clear now that there is a range of ways to embed analytics within operational applications and processes. Each approach makes BI easier to use, more flexible, more pervasive, and more actionable to varying degrees. There are also many different ways to develop embedded applications. These development methods often parallel the architectures listed earlier, and we have hinted at some of them. But it is worth examining these methods in more detail.

1. Custom Coding. Developers can hand-code BI functionality from scratch or embed the code of open source BI tools within their existing code. This parallels the "local services" option described in the previous section.

2. Integration Services. Developers can also write calls to remote BI servers using a BI tool's software development kit (SDK) and other APIs offered by the vendor, such as portlets, Web services, Microsoft Smart Clients, and so on. While integrating disparate applications has gotten simpler over the years thanks to the Web, it always requires some degree of custom coding. And the more tightly integrated the applications, the more coding is required.

For example, developers can integrate BI products to various degrees, depending on business requirements and the richness of the APIs provided. Portlets, for example, generally provide lightweight integration. The portlet connections are often brittle because they don't support sophisticated process handling conventions such as error recovery, error messaging, timing, and message queuing. But these types of connections are often sufficient for most intranet portals that provide supplementary or non-mission critical information to employees.

Tight integration between BI and other applications requires lots of custom development.

However, other types of portal applications may have more stringent application requirements. For example, a financial services company that wants to give customers a secure way to examine and drill into their 401(k) statements over the Web needs to create an extremely tight connection between its customer-facing portal and its interactive reporting engine. Likewise, a commercial software vendor that wants to deliver a suite of analytically driven business applications comprising products it purchased through an acquisition or licensed in an OEM deal needs to rewrite the applications to run on a common architecture with standard services for security, metadata, file formats, and object models.

3. Component-based Development. The most promising method of embedding analytics is to use emerging software development workbenches that make BI

services available to developers as components that they can drag and drop onto a screen, configure, and link together. These tools not only simplify and accelerate the development, testing, and deployment of analytically driven applications, but they also use BI components, which fosters reuse and standardizes the look/feel and delivery of BI functionality within an organization. Development groups that want to reduce backlogs and be more responsive to business needs should employ these approaches. Two types of workbenches are available for the analytically minded developer: IDEs and ADEs.

IDEs. Traditional interactive development environments or IDEs are offered by commercial software vendors such as IBM (Rationale), Microsoft (Visual Studio.NET), the Eclipse Foundation (Eclipse Project), Borland (JBuilder), and SAP (Visual Composer). Many of these IDEs have added BI components to their workbench palettes so developers can embed BI functions within non-analytical applications. These components must still be tied to analytical engines residing either within the local application server or remotely on another server.

Most ADEs eliminate the need for coding altogether.

ADEs. Some BI vendors offer developer workbenches devoted exclusively to building analytic applications (collections of interactive reports and views designed to support specific business tasks and processes). These analytic development environments, or ADEs, work the same way as IDEs (hence the copycat acronym), but often abstract the development process to a higher level so that power users, not just developers, can rapidly prototype and build applications with the toolsets. Most of these ADEs eliminate the need for coding altogether when building BI solutions.

The first vendor to offer an ADE was Alphablox, which was founded in 1996 and is now owned by IBM. (See Figure 5 for a screen shot.) Only in the past few years have other vendors begun offering ADE workbenches. These include Microsoft's BI Development Workbench, Business Objects' Performance Management, Board MIT's Management Intelligence Toolkit, and arcplan's dynaSight. The more mature solutions enable thousands of users to tailor their data to meet their individual needs and roles.

Alphablox's Authoring Environment

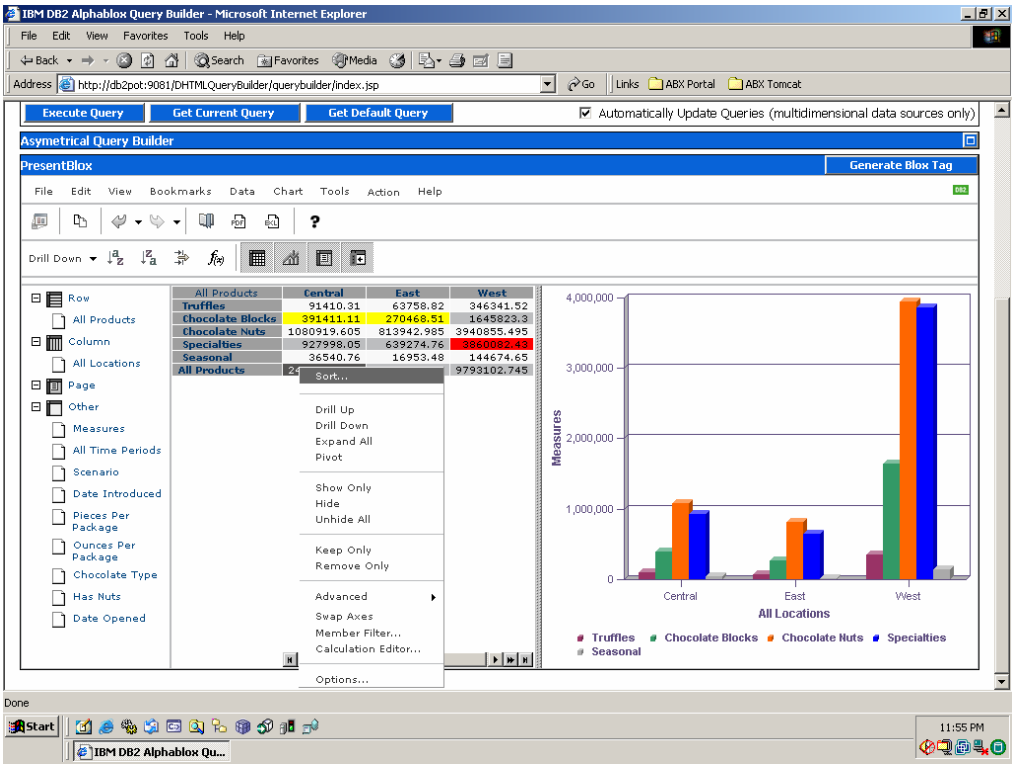

Figure 6

Many report authoring tools offer much greater flexibility than in the past and mimic many features of ADEs. For example, many reporting and dashboard tools now use a desktop publishing paradigm with drag-and-drop functionality, which enables power users to create rich displays of interactive reports. However, few of these report design environments make it easy to execute processes in other applications or handle requests for data from other applications. The best ADEs provide a high degree of flexibility while keeping complexity at a minimum.

Conclusion

The Value of Embedded Analytics. Embedded analytics bring BI closer to the operational processes that drive business activity on a daily basis. Embedded analytics make BI more pervasive, because they deliver information and insights to all users within the context of the applications and processes that they use to do their jobs. In addition, by more tightly coupling operational and analytical functions, embedded analytics give users access to timely information so they can monitor business events closely and work proactively. Moreover, embedded analytics can kick off automated actions as conditions change to optimize business processes and performance.

Approaches to Embedded Analytics. Given such benefits, it's not surprising that there is a range of ways to embed analytics within operational applications and

business processes. Organizations can embed BI functions and services locally, call them on remote servers, and integrate BI modules within an application suite. They can also use BI services to trigger alerts, workflows, or executables in other applications. To develop embedded analytics, developers can create BI functionality from scratch, call BI services on local or remote servers, or use application development workbenches to embed and configure BI components within an operational application.

Embedded analytics are ultimately the way that BI will become a mission-critical resource within all organizations.

The Future of Embedded Analytics. Embedded analytics will not replace stand-alone BI tools. Rather, embedded analytics will make the functionality offered by such toolsets more readily available. By embedding BI functionality within operational applications and processes that drive the business, embedded analytics will make BI more operational, easier to use, and pervasive—key challenges facing the current generation of BI adopters. Embedded analytics are ultimately the way that BI will become a mission-critical resource within all organizations.