
Corporate Portals *Letter*

Industry exec envisions next step for portals: e-analytics

The relationship between portals and e-analytics, rules-based programs that filter data to present relevant information to a particular audience in real time, has traditionally consisted of either personalization tools or specialized applications. But today, some vendors are taking the relationship a step further.

Currently, almost every portal vendor makes it possible to personalize the interface by role, individual preferences, and rights, to filter down the massive amounts of available data to a more meaningful picture. In addition, business intelligence specialists bring a powerful analytic engine, specialized reporting, and decision-making applications to the portal interface.

Now, some enterprising vendors are working to make e-analytics a core functionality of the portal. This development holds promise for organizations seeking to use an enterprise portal for mission-critical functions. “If you look at areas in the organization where the premium on effective decisions is high, and the ability to make those decisions is based on lots of variables that are changing in real time, this is where e-analytics is going to be effective,” says analyst Nat Palmer of the Delphi Group.

One pioneer in this field is Verilytics, Inc. Formerly known as iBelong, Verilytics recently acquired analytics firm Redwood Investment Systems, and is incorporating Redwood’s technology to create a portal platform that gathers, analyzes, and delivers highly targeted information to users in real time—providing, as Palmer explains it, “contextualized answers rather than simply filtered information.” The new platform will be released later this quarter.

For organizations and portal architects, what are the implications of this development in the portal space? To find out, *Corporate Portals Letter* spoke with Verilytics founder and CEO Shikhar Ghosh. Below are excerpts from the interview.

CPL: First of all, how do you define e-analytics as it relates to portals?

Shikhar Ghosh: The first notion of the portal was that it gave you a single interface into a whole lot of sources of information, typically hosted in different places on different systems. The next version that came along allowed you to personalize those sources of information. E-analytics is a continuation of that progression. It starts with the premise that there’s way too much information, and most of the information, most of the time, is not useful. E-analytics creates a whole layer of applications between the information and the user that distills that information down, based on a set of rules or guidelines or agents, to bring only that information that’s really useful.

For example, most portals have some level of stock indicator. The first generation of portals lets me watch what was going on in the stock market. The next generation lets me watch just what’s going on in my own portfolio. Where we’re going with e-analytics is to say, “I don’t even have time to watch my own portfolio all day. So when something really interesting happens, then let me know. And at that time, let me go very deep into that information.”

Volume 2, Issue 3
March, 2001

INSIDE:

Web Site Roundup	4
Product Spotlight: Sybase	6
Glossary of Terms	5
Case Study 1: MetricStream	8
Portal Events Calendar	10
Case Study 2: Top Tier.....	12

How does this differ from personalization?

The personalization screen on most portals allows you to essentially just use the minus sign or delete column of a portal, where you specify, “I don’t want the 10,000 stocks, I only want these five.” What it doesn’t allow you to do is to combine information, so that if things exceed an average, you’re notified.

In many ways, the current portals look a lot like Microsoft Word. You can create something, you can manipulate it, you can change fonts, you can do a lot. But what we’re trying to create is Microsoft Excel, in which you can actually combine things and create new values, new information out of what you had before, and apply a set of rules to it.

Would it be a correct extension of the analogy to compare the built-in functions in Excel to the CDAs or gadgets in portals today, the programs that you use to sound an alert when inventory items get below a certain level?

I think that’s the first step of the analogy, but if you think about a spreadsheet, what you’re doing there is combining two different kinds of information to yield something new. If I’m interested in the price of Cisco as a stock I might say, “When Cisco exceeds 25, let me know.” The real problem is that rule becomes invalid once Cisco’s at 27, because then it’s always exceeding 25. Then I’ve got to go back in and change it to, “Let me know when it exceeds 28.” What you really want to have happen is that when Cisco exceeds the average for the last five days, or when Cisco goes up, and the average of other tech stocks goes down, you’re notified. So there’s a lot of computation going on well before that rule gets applied.

And you can extend that to a lot of things. A purchasing manager who’s buying from a business exchange doesn’t have time to keep watching the exchange and seeing what the different prices are that he’s interested in, whether they have enough quantities, and so on. So he can put his buy list in there and specify, “If any of these items is more than 20% off what I normally pay for it, the quantity is more than a certain amount, the delivery date is within the next three months, or if the total of all of these (because the whole market’s dropped) allows me to save more than \$10,000, put up an alert.” And it’s that combination of two or three pieces of information that often come from different sources that make the analytical part valuable. It’s really the next step to personalization.

The product you’re working on personalizes data at the domain level. On the technology side, how does this work?

There are three premises on which we work. First, we start by defining all the data elements from the user’s perspective. For example, if the user is a purchasing manager, then their purchase list is an element—all the items they buy, in the forms they buy it. If you’re a stock analyst, an element might be a portfolio. So we start by defining all the names within the system in a user-centric way. We identify all the dimensions of each particular element. If I take IBM as a stock, that has many different properties. IBM is a large capital company, it’s a high-tech company, it has operations in Eastern Europe, it has operations in Southeast Asia, there are many dimensions to it. So you attach all those dimensions to the name. And within that naming, domain knowledge becomes really important. If the user is thinking finance, the sorts of things that are important to them are very different than the user who’s thinking sports.

The second premise is that we’ve divided the system up into three separate layers. The first layer is content rendering and personalization, which most portals have within them. The second layer is the analytical layer—sorting the information based on a set of rules. It’s about creating norms. It establishes the normal things you expect from this sort of information. For example, say the user is expecting five stories a day on Cisco. If it’s just five stories, if the user wants to go in and look at the stories he can. But if it’s twice the norm, then something’s going on with Cisco and the user should be alerted. So it’s constantly measuring things against some level of norm. It’s trying to



Shikhar Ghosh, CEO, president, and founder of Verilytics, Inc.

figure out what events and deviations are occurring. The third layer in the system is mapping how the user uses the information to how the information systems have it displayed. So that if you change an information source at the backend, you're not affecting any of the user functions. A lot of that layer we do in memory.

The third premise is that because portals and the way we've defined e-analytics are a large integration function, we've been scrupulous about everything being done in standards. So the whole structure, all of the interfaces, are all J2EE-compliant, 100 percent Java-based. We're moving everything into XML. Otherwise the integration task becomes huge. In many ways, you couldn't do this two years ago, because XML and Java weren't mature enough to deal with it.

For an organization planning or implementing a portal, what's your recommendation for what they should look for now, given what might be available six months down the road?

A starting point for an organization is to ask, "How do I want as many users as I possibly can—some of which are inside my company, some of which might be my partners or my customers—to use the information that I have?" And then, "What kind of systems do I need to integrate?" In most cases, the cost of integration is greater than the cost of the systems or the cost of the information. And to start with the premise that what you really want to do is to integrate at the user level and bring in as much information as you can. That has a much greater return on investment than building one more system in the back that meets some specialized need.

What's starting to happen with Java and XML as technologies is we're starting to get the ability to combine vast amounts of data from very disparate backend systems into a common framework. But you don't want to take all that information and give it to your user directly. Because most likely what the user will do is put it in a spreadsheet or just focus on one piece of data. You want to look at how the user is going to use the information. Let me process those things that computers are good at processing, and give the user something they can directly use. And the closer you can get to information that the user can directly use, the more productive the user will be.

Can you give an example?

If you're a user and you're looking at what your risk exposure is, you might want to know, "How exposed am I to a change in the economy in Southeast Asia?" There, what you want is a combination of how much cash you have out there, what your total investment is there, how much equity you have there, all different kinds of instruments that might come from different systems. But the thing that you're really interested in is your level of exposure.

If you took all the systems on your backend and provided all that information to the user, the user would have to take all those pieces of information and put them onto a spreadsheet, and then calculate their exposure. And the user couldn't do that every day, every minute, whereas those are the sort of things that computer systems do really well. So you start with the notion of exposure, and the way this particular user defines exposure is equal to their cash exposure, their equity exposure, and their currency exposure.

When he has that exposure number, typically the user will ignore it most of the time. But when it hits a certain threshold, the user is going to get really interested in it. And at that point, what you want the user to be able to say is, "My exposure's just exceeded some threshold, now I want to know everything—what are the news stories, what's been written, why has this happened?" And you want to make all of that directly available to the user.

That combination of distilling it down to just the essence of what the user needs, but at the time it hits the threshold—going back and making everything that can be discovered in your systems available—that's a hard problem.

The problem used to be, "How do I take all this information and manage it?" Then the problem got to be, "How do I integrate my systems?" Those problems have largely been solved from a technological point of view. The problem that's much more difficult to solve and the one on the cutting edge is, "How do I do a user-based integration of all the things that I have?"

A user-based integration?

If you think about the world from a data perspective, and of bringing together of data, you would call IBM's price a piece of data. Then you take it one step further and say, "Here's a news story that happens to

Glossary of Terms

Alert – Action triggered by an event occurring or threshold reached in the system; can take the form of an email message, page, or execution of a program.

E-analytics – Rules-based programs that filter data to present particular information to a particular audience in real time.

Exception – A condition that causes a program to take a specific action.

Metrics – Centrally-defined queries into one or more data stream, to produce analytical results.

be related to IBM.” Whereas a user looks at it as, “I want to know everything about IBM.” That includes stock price, what happened in the news, and so on. But if you take the user’s perspective one step further, what he really wants to know is what happened to his portfolio, and IBM’s just one element in it. He’s really interested in the whole thing as a whole. “What’s my risk in it? Has it gone up? Has it gone down?” You take it one step further and what the user really wants to know is, “What’s my wealth?” Which includes their equity portfolio, but also how much cash they have and a bunch of other things.

And when you start from that perspective, you’re typically creating aggregates of different kinds of data. An aggregate of data would be wealth, the next aggregate would be my portfolio, the next aggregate would be the stock price. At that point you’re starting to bring together different kinds of information. Then, when something unusual happens, you can then drill down and see what’s going on. Otherwise, for an average user it’s almost impossible to keep track.

E-analytics sounds like it can improve productivity in the portal. Do you see challenges to getting end-users to take advantage of the functionality?

You hear things like, “The technology is great but getting the users to change how they do business every day is tough.” Typically, when I’ve made the same statement, it’s because I’m frustrated that users are not using what I want them to use. This is true not only of portals but of many kinds of technologies. They’re created by technologists who think about what a user *should* want. They have all this information and they want to present it because it should be good for the user.

In fact, users don’t have the time or ability to process all that information. They have certain norms and rules that they follow. We’re trying to start by asking how users use information. “What are the things they actually do? What are the concepts they actually have?” And then build a system around that.

You almost start by peering over their shoulders and seeing what they’ve written on their pads and what they have on their spreadsheets. And then you figure out how to make that available to them directly and on their terms, in the concepts that they measure. That’s when the use index goes up dramatically, because now you’re starting on the user’s terms instead of the system’s terms.