

TrendMarkers

technical trends ■ product reviews ■ performance benchmarks ■ application server market

IN THIS ISSUE

Editorial: Notes From the JavaOne Conference	1
Dotcoms: Blood Bath or Reality Check?	2
Why Use Free Software?	5
What's New in PHP4?	8
A Close Look Under the Hood of B2B Solutions	15
Application Servers: Gem-Stone Swallowed Up by Brokat	19

TM READER REACTION

If you have something to say about one of the articles appearing in TrendMarkers, then send us a letter at: <http://techmetrix.com/lab/trendmarkers/tmksendletter.shtml>

July/August 2000
Volume 2, Number 7/8

Editorial: Notes From the JavaOne Conference



By Jean-Christophe Cimetiere – CEO
jcc@techmetrix.com

The fifth edition of JavaOne was held June 6-9, 2000 at the Moscone Center in San Francisco, CA. According to Sun officials, the event was even more successful than had been anticipated.

While there was certainly a lot of hype surrounding some pretty cool innovations, most of the keynote addresses seemed to focus on the same topics: money and business.

Both Sun and guest speakers tried to spark the crowd's interest by asserting that Java would help them make some big bucks. For example, Steve Jobs dropped in long enough to let folks know about Apple's commitment to the Java platform, and another Apple source said that WebObjects would abandon Objective C support in favor of Java support. The audience's reception was surprisingly cool, but certainly realistic. Long gone are the days when someone opened a Web site and became a millionaire a short time later. Venture capitalists now want to get their money back (see the article, "Dotcoms: Blood bath or Reality Check?").

Beyond all the demagogical big money talk, attendees were obviously at the convention to find real solutions for real businesses. However, the event was not entirely void of flashy demonstrations.

Quite a few Java-embedded devices were on display: PDAs, phones, car systems, game consoles, interactive TVs and so on. "Get connected anywhere, anytime" seemed to be everyone's slogan. In listening to the vendors one would think that every possible device must be wireless. Hold on there, guys. Remember that wireless is only a very small part of most companies' e-business strategy (see "Is Your E-business Strategy Ready for Wireless?" <http://www.techmetrix.com/lab/trendmarkers/tmk0400/tmk0400-3.shtml>). Even though today's most advanced applications on Internet-wireless enabled devices (like the Palm VII) are interesting, they are still very limited.

I particularly appreciated the concepts demonstrated by Zucotto, which has brought together Java and Bluetooth using a dedicated Java chip and a toolset. This makes it possible to build Java-Bluetooth appliances and then easily develop the required software. But Java has not yet become the leader in embedded devices. There are still many competitors out there: Palm OS, Windows

(Continued on page 2)

Editorial: Notes From the JavaOne Conference (cont.)

(Continued from page 1)

Embedded, Linux and all manufacturers' proprietary systems.

Business-to-Business (B2B) was another important topic addressed at JavaOne. Even though it was a Java event, XML was the star of the show for B2B solution providers. B2B is currently fueled by many technologies including, but not limited to, Java. For example:

- Middleware technologies (message-oriented, ERP connectors, system gateways, legacy bridges)
- J2EE-based and Microsoft DNA application servers

- Automated business processes and rules engines
- XML data conversion/transformation, XML data mapping, XML standard schemas, XML schema translation
- Security management

If you bring together all of the above, you've got yourself one killer B2B solution. Several vendors are currently working hard to develop such a product. This market is a mix of old (middleware veterans) and new players and has more and more to offer each day. While it is good to be spoiled for choice, such a situation makes it quite difficult to select a product. In this issue, we'll look at both XML and B2B in hopes of clarifying available and emerging solutions. ■

Dotcoms: Blood Bath or Reality Check?

By Alain Lefebvre – Vice President of Groupe SQLI
alefebvre@sqli.com

The prevailing feeling in the U.S. regarding dotcoms has come full circle: a year ago, start-ups were re-inventing the economy, wooing investors and attracting piles of money. Today, skepticism reigns and bankruptcies are multiplying at a fast clip.

What has changed over the past year?

Not the economic context or technology. Not advertising, where money keeps pouring in. People continue to use the Web in ever greater numbers, millions of new users joining those already on line.

Yet the hysteria that led to the "anything goes" period has given way to mistrust and the money tap has dried up. Palpable in the U.S., the crisis has not yet crossed the Atlantic. The time it will take for it to reach Europe will be a valuable indicator of the gap remaining (at the "Internet experience" level) between here and there.

Sites are being closed in all sectors: gen-

eral public and B2B, commercial and content sites.

Not all sites in trouble are closing down but downsizing and consolidation (pets.com absorbing petstore.com, for example) activity is a good indicator of the depth of the malaise. Reasons are diverse and the lessons to be learned interesting (even if they only confirm known principles):

- Lack of usability (didn't understand the Web and its interface)
- Wrong economic model (not enough income)
- Wrong business model (expenses too high)
- Wrong investors/shareholders (no time to prove the concept's validity)
- "Successful" combination of several of these mistakes (or of all of them if companies are high performers!)

Curiously, very few sites fall due to lack of usability; so far only Boo.com ranks in this category.

(Continued on page 3)

Dotcoms: Blood Bath or Reality Check? (cont.)

(Continued from page 2)

Boo.com offered a flashy home page that took a long time to load, was inaccessible to Mac users, was hard to comprehend and even harder to use. If sales were low, it's simply because customers were UNABLE to find the items or, if they did, were unable to complete their order! Here's an e-commerce site that didn't work, that broke every usability rule. It (quickly) died and got what it deserved.

Reel.com is a perfect example of a well-known site that attracts a lot of traffic but does not know how to take advantage of its popularity (lack of economic model). Reel.com was launched in September 1996 and thus was not a latecomer. In the beginning, it offered reviews and a complete list of films playing in theaters, and its rich content attracted visitors. But instead of leveraging this attractive package (and selling ensuing traffic to advertisers), Reel.com attempted to morph into an on-line video store. This was not a profitable venture because of price pressure from the likes of Amazon.com and buys.com. Principal shareholder Hollywood Entertainment Corp. preferred to throw in the towel rather than keep sponging the deficit.

All failing sites that crumble under the weight of their expenses (the famous "burn rate", the speed at which you "burn" money...) are not victims of bad economic models but rather unsuitable business models.

Indeed, financing through advertising keeps many – and not only large – sites alive. You simply have to line up income potential and investment capacity (or accept to depend on the good will of investors and/or the stock market; this is essentially the type of reversal we are witnessing today!). It's the lesson Salon.com, an e-zine that was a new media darling and is now having to lay-off a great number of employees, is learning (and so are cbs.com and worse, abpnews.com, which is shutting down)...

These "new journalism" sites are mainly

victims of their own delusions of grandeur: was it really reasonable to hire 150 people to keep Salon going?

Ken Layne, co-founder of tabloid.net, thinks "the lesson to learn from these collapses is that there is no need for 150 employees to keep a site going. It's ridiculous for Salon to have gone through \$150 million last year when it had none of the burdens of the written press, no printing and no distribution costs" as quoted in the New York Times.

The rules have changed: yesterday (a year ago), the market was clamoring for expansion and [very] high growth. Today, the mantra has become "profitability first!"

Entrepreneurs are disoriented by this change in attitude. As Mark Sauter, CEO of APBNews, so aptly puts it, "until Nasdaq fell in April, the market was telling us that we had to become an international multimedia corporation. In 72 hours, the message had switched. All of a sudden, the market was telling us that we now had to build a modest site and turn a profit in the near future. We find ourselves at the wheel of a semi-trailer on a mountain road in Montana, having to slam on the breaks in a hairpin turn." (New York Times)

Those who spent money as if there were no tomorrow to get known are having trouble but they aren't the only ones. There are also the start-ups that accepted to come under the wing of a large shareholder when perspectives were rosy; now that the wind is turning, attitudes are changing and the guillotine is falling. It's the sad tale of Toysmart. Com.

Toysmart management accepted a Disney buy out a year ago, at a time when large corporations were saying, "quick, we need to move into e-commerce." Disney recently reviewed its Internet strategy and decided its Toysmart investment was no longer strategic... Why continue to support a high burn rate when the sector (selling toys on-line) was crowded and clear winners were emerging (chiefly Etoys and Toysrus)?

(Continued on page 4)

Dotcoms: Blood Bath or Reality Check? (cont.)

(Continued from page 3)

And so they shut down Toysmart and moved on to other things (after all, founders and employees should still be able to easily find other jobs and better sooner than later.)

This type of failure (mainly due to the difficulty of continuing to finance losses) will multiply too as it becomes apparent that investments are what they've been since the very beginning: hazardous bets (difficult to justify in the long run).

Finally, there are the "unforgivable" cases. Drkoop (a medical portal site), for instance, is symptomatic of dotcom fever excesses. Launched very quickly on Nasdaq (with no income and no history) by avid bankers, the money raised was nearly instantaneously totally wasted through outrageous agreements with AOL. The income potential soon appeared as it really was: limited while marketing expenses were limitless.

Drkoop has not yet fallen but its life expectancy has shot down (due to absurd management decisions), it is now at most only a question of months. Drkoop is the perfect example of what shouldn't have been but

was done by so-called responsible professionals:

1. Bankers shouldn't have promoted an "empty" value.
2. (Hallucinating) marketing agreements should have elicited more critical response from analysts.
3. Finally, we shouldn't have waited for accountants to sound the alarm before grasping that Drkoop's burn rate was sending it straight into the wall.

But excesses did occur. There are many other Drkoops in the making, the dotcom bloodbath is barely starting. This being said, this purge, as unpleasant as it may be, is necessary and it will also happen in Europe (it has already started with Stoodly, a student-oriented portal). Here too, you have to keep your wits: e-commerce and content sites will not suddenly reveal themselves as a no man's land to be shunned at any cost. Beside giants such as Amazon, Yahoo and the Wall Street Journal, specialized and even "proximity" sites remain largely to be invented and winners should still be plentiful (while being cheaper to set up). But clearly, the pendulum is swinging back, a warranted occurrence for some but a little unfair for many others. ■

About the author

Alain Lefebvre is co-founder and vice president of Groupe SQLI (<http://www.sqli.com>), a total-Internet agency specialized in defining and activating a useful and effective Internet presence for its clients. A major new technology player for more than a decade, Groupe SQLI offers full service and continuing coaching to enable companies to move towards profitable total-Internet solutions.

Author of three books dedicated to client/server and intranet, Mr. Lefebvre is also a regular contributor to many specialized publications. His upcoming book takes a close look at the strategies used by today's most well-known Web sites.

Why Use Free Software?

By Alain Lefebvre – Vice President of Groupe SQLI
alefebvre@sqli.com



The current surge towards Linux has greatly contributed to popularizing the idea of free software. Today, all of the major players are speaking in favor of this system (in 1999, Linux represented 25% of the server systems sold). Even Microsoft has admitted in its own way the importance of Linux and OSS by criticizing them profusely on its Web site!

Linux is not the only system to benefit from such a craze. Other software derived from open projects is also beginning to enjoy the kind of recognition such distribution deserves (Apache represents 60% of HTTP servers on the Web, Sendmail makes up 40% of the SMTP servers on the Internet, etc.). This wave has begun to reach enterprises and IT department heads are rightly beginning to think about using Linux and the software surrounding it. In this context, questions (often tainted by skepticism) abound:

- Does free software perform well?
- Is free software reliable?
- What about follow-up and technical support?
- Are there any guarantees in terms of evolution?
- Free software is available at no cost... but for how long?
- Can someone retrieve free software (and at the same time the user base)?

The goal of this article is to respond objectively to these important and legitimate questions.

Response # 1: Free software is not a fad.

Virtually unknown outside a closed circle of specialists a mere two years ago, the stars of the OSS movement such as Linux, Apache and Perl have been in the headlines over the past few months – and not only in the specialized press! The computer world is no stranger to such sudden crazes, which can make one believe the worst: a

smokescreen of some techies in need of publicity, a new anti-Microsoft crusade, etc. In reality, Linux (and other free software) is interesting for other reasons than the fact that it isn't property of Microsoft.

Linux proves to be the "unified Unix" that the industry had been waiting for for ten years. It enjoys qualitative contributions from both the Unix and OSS worlds. Moreover, the movement towards Linux has become so powerful that has even pulled in those such as SCO and Novell that have the most to fear from its emergence.

Response #2: Linux is quick.

In considering how Linux runs we obtain a global response to questions about the quality of all OSS projects because this system, which has become very popular as of late, can be perceived as a "concentration" of free software. Indeed, the Linux kernel is completed by a compilation of the major OSS projects: Apache, Sendmail, Perl, Xfree and so on. A recent "TPC-D" bench pitted Linux against Windows NT using Oracle 8 as database for the two systems. The results, which were published on the Internet, showed that Linux was superior by a ratio of 2 to 20. The test conditions were voluntarily simplified: common server configurations were used (Dell Power Edge), as were default software parameters. The objective was to measure the potential of each when implemented by non-specialists (which is typically the case in the small and medium-sized businesses targeted by these environments).

One could always argue that Windows NT, as well as Oracle 8, is very sensitive to tuning and that some additional tuning could have considerably changed the results. This is certain, but the same argument could be made for Linux.

How can we explain that Linux proved much quicker than NT to run Oracle 8? First of all, Oracle developers had access to a larger amount of information concerning

(Continued on page 6)

Why Use Free Software? (cont.)

(Continued from page 5)

the internal characteristics of Linux than they had concerning NT; porting takes place in the very best conditions and this is felt when it runs. Next, it is patently obvious that with equivalent machines, Linux uses fewer resources than Windows NT. This is simply due to the fact that Linux is truly modular. It is equipped with a clean architecture and does not impose a manner in which programs must be run. The graphical interface is free and based on X-Window, which has been a tried-and-true standard in the Linux market for the past several years. But most importantly, the latter is not loaded automatically at start-up, which avoids useless consumption of memory when running a dedicated server. On the contrary, Windows NT launches its graphical interface (GDI module) automatically and without exception as soon as it is booted, and this is unavoidable. More recently (September 1999), Siemens published the results of a benchmark carried out with SAP R/3 Release 4 within which Linux once again proved quicker than all other operating systems.

Response #3: Free Software is Reliable.

The Web is overflowing with testimonies of free software's proven reliability. Free software programs are often used for critical applications and much to the satisfaction of their administrators, they run in a "boot and forget" manner. This is why we often find Apache, Perl, PHP, Sendmail, Linux, FreeBSD and others at the heart of the configuration of the largest commercial Web sites (Yahoo, Hotmail, etc.) for which availability is a major concern. Another domain in which high reliability is crucial: embedded information systems equipment. Here again, we often find Linux. Also, computer security specialists often put their confidence in Linux for implementing firewalls, mostly because it offers open code that has been very "reviewed and revised." What's more, improvements are made available as soon as a glitch is discovered.

Why is OSS reputed for its good quality while proprietary software programs suffer

more and more from a lack of quality that hurts their image? (This is perfectly illustrated by the 1998 decision of the Web BugNet site, which is specialized, as its name implies, in information about program bugs, to no longer award its annual prize for best program correction). The answer is most certainly the Darwinian approach to development that distinguishes OSS projects and encourages numerous parallel developments (something which a traditional vendor would never do due to a lack of resources and fear of conflict between development teams). In the end, only the "kernel" group retains the best contributions. But even more so than during initial development, it is during the fine-tuning stage that the multiple interventions characteristic of OSS have the biggest impact on final quality. Debugging is particularly effective due to the idea of "strength in numbers;" there is a far greater number of testers than developers, and such exposure results in a level of corrections/improvements that is completely foreign to proprietary projects that are governed above all by a scheduled launch date.

Response #4: OSS is supported.

The problem of technical support can alone put a damper on the zeal for free software. It is generally believed that there is simply no technical support for this type of software. This perception is far from true. InfoWorld even presented the Linux community with an award for best technical support in 1998. OSS user and developer communities are known for being very reactive. In most cases, soon after a bug is identified, the corresponding correction is put online. No matter how effective this system may be, decision-makers just can't seem to make do with this system — maybe for psychological reasons!

This explains why there has been a recent surge in commercial service offers surrounding OSS. First of all, it is quite natural that the vendors circulating Linux in the form of distributions also offer training services, hot lines, etc. However, we have also witnessed the birth of companies created

(Continued on page 7)

Why Use Free Software? (cont.)

(Continued from page 6)

especially to meet this demand, for example, LinuxCare. Other specialized companies propose technical support contracts focused on Sendmail or Apache. These contracts propose both on-site interventions and telemaintenance (this type of direct—but remote—intervention by telephone is common in the Unix world but still impossible with Windows NT). Even more significant, the giants HP and IBM now handle Linux technical support that is available throughout the world, 24/7. With such a wide range of choices, the question of technical support today boils down to choosing the type of support you prefer.

Response #5: How will OSS develop?

Behind this question lies all the uncertainty (or even worry) that collaborative development triggers in the minds of decision-makers. This question also reveals a deep misunderstanding of those involved in the OSS movement. It is true that if those contributing to OSS project did so merely out of the kindness of their hearts, there would be reason to doubt how long the results would last. In fact, most of the developers who actively contribute to the evolution of Apache, PHP or Linux are directly concerned and capitalize on the popularity of the software on which they work voluntarily.

As an example: the vendor Red Hat assigned three of its employees to Linux kernel development. In doing so, Red Hat was not being altruistic — just looking out for its own good. Indeed, the more the kernel evolves, the more Linux will be distributed, and the more potential clients there will be for Red Hat products and services.

Companies specialized in Web development that have built their reputation on their excellent command of Apache, Perl or PHP react in the same way for similar reasons. OSS is one example of a mutualistic principle in which the participants take away more than they give (Internet is the other famous example).

Another objective argument rallies in favor

of OSS' longer durability compared with that of proprietary software programs: since OSS is not (and cannot) be controlled by a public or private party, projects do not risk to be interrupted for political or marketing reasons. On the contrary, the proprietary sector is accustomed to such problems; for example, when Sun decided to transfer its clients from SunOS to Solaris.

Response #6: For the moment OSS is available at no cost... but for how long?

First let's look at the concept of "available at no charge". Open Source programs are distributed for free and will remain as such. Even vendors such as Red Hat that offer "packaged" Linux distributions on CDs with documentation make it possible to download a complete version from their FTP servers at no cost. This is quite simply because the GPL license on which most of these projects are based requires such behavior.

Free-of-charge distribution has not hindered the emergence of a different, profitable market, quite the contrary. The famous messaging software Sendmail (available with its source code and at no cost) is also commercialized in a "more accompanied" version by a company created by the original developer of Sendmail. This version (Sendmail Pro) is not a divergence of Sendmail, which continues to evolve and remains the basis of the "Pro" version. Sendmail is only one example that proves it is possible to create a mutually beneficial relationship between an OSS development and an added-value commercial product.

However, it is important to understand (in the case of Sendmail and similar cases) that even the original developer/creator of Sendmail (Eric Allman) was only able to make money off of his creation by creating a company that is on the fringe of the movement and without ever pretending to control, manager or reclaim the rights to his creation. On the contrary, it is still in his best interest that the free version of Sendmail be widely distributed (and thus regularly enhanced and improved) as it widens his market for spin-offs. ■

What's New in PHP4?

**This is a follow-up to last month's article, "Free Software Profile: PHP," which did not deal with PHP4's new features.*

By Christophe Lauer – Head of R&D
clauer@techmetrix.net



The long-awaited PHP4 version has been available since the end of May 2000. After having acquired successively Beta-test and Release Candidate status, the final version which can be used for real-world site production is today a reality.

The questions that come up systematically with each new version of a software program are, 'What's new in this version?' and 'Which of these new features are truly useful?' These are certainly legitimate questions, as the simple availability of a new version does not justify migration to it. The thing is that behind what seems to be a simple update looms a myriad of potential problems that must be zeroed in on before moving forward, especially because once updated, there's often no turning back.

I'd like to take a minute to point out that if you are currently questioning the availability date of PHP4 support by your favorite host, it is important to know that such a task represents a considerable amount of work.

In this article, we'll take a quick look at PHP4's new features, which will allow you to appreciate the tremendous work that has been carried out by Zeev Suraski and Andi Gutmans, as well as all of the other contributors.

What's the difference between PHP3 and PHP4?

PHP4 is a complete rewrite of PHP3 at the heart of the language's engine. But why did the language's authors decide to completely rewrite it?

The manner in which PHP3's scripting engine works is not adapted to the execution of rich, large-scale applications such as Phorum (<http://www.phorum.org>) or KeyStone (<http://www.stonekeep.com/keystone.php3>). Indeed, this engine reads PHP script instructions line-by-line and

then executes them one-by-one, which enables PHP3 to obtain good performance with short and simple scripts, but yields rather mediocre performance with larger applications. It is important to understand that PHP3 was not designed to execute such applications.

It is for this reason that PHP3's two creators decided to implement a scripting engine that would be capable of reading all PHP code and "compiling" it before executing it: Zend. This was not an easy task. Although incompatibilities could be conceived at the time of migration from PHP2 to PHP3, such differences had to be limited as much as possible between PHP3 and PHP4 due to the installed base and the number of PHP3 scripts already in use. We can congratulate PHP4's authors on a job well done because the incompatibilities that do exist are only minor.

Although Zend is an integral part of PHP4, it is in fact a scripting/runtime engine that is independent of the PHP language. On this account, it is possible to use Zend in MySQL as the basis for the development of a stored procedure language specific to this database system.

Without going into the details of internal implementation, we can point out a few things about the Zend/PHP4 couple.

PHP4's new architecture allowed its authors to add an abstraction layer in relation to the Web server at the heart of its language. Up until now with PHP3, we had the choice between using it as a specific module for the Apache server or in CGI mode for all other HTTP servers. With PHP4 it is now possible to imagine better integration with Web servers other than Apache. In fact, this is the case for Microsoft's HTTP server, IIS. PHP4 now works as an ISAPI filter for Microsoft's IIS server. For this reason, different parts of PHP4 code had to be made compliant with use in a multi-thread environment (PHP4 is thread-safe").

(Continued on page 9)

What's New in PHP4? (cont.)

(Continued from page 8)

There are many new features concerning PHP4's internal implementation, but they don't really concern PHP programmers directly. Nonetheless, it is worth noting one of the most important points: memory allocation and reference counting mechanisms, in other words the mechanisms that free up resources which were no longer used, were completely rethought. Performance and memory use are thus improved, especially for objects and tables. The more data the variables contain, the more memory is saved.

A Developer's View

What are the major differences between PHP3 and PHP4 from a user's (a PHP4 designer/developer) point of view?

Sessions

One of PHP4's most anticipated functions is native session support. Indeed, nothing in the language's base functions provided for management of user sessions, in other words, all of the values and variables that are linked to a user and can be transferred from page to page within the same application.

With PHP3, it was necessary to build a session system manually by using cookies, by making an identifier a parameter, or by using PHPLib. PHPLib can still be used with PHP4 because it offers other functionalities besides session management.

In PHP4, a whole group of configuration features and directives in the `php.ini` file take the burden of session and user context management off of the programmer. PHP4 session variables can be stored in a simple ASCII file or in a relational database. All of these features are now documented in the official PHP manual (<http://www.php.net/manual>).

Output Buffering

With PHP3 all of the content produced by `echo ()` or `print()` was directly sent to the server. In PHP4 an intermediate storage

layer makes it possible to defer the sending the Web server's output flow. In order to do this, new features, which haven't been documented yet, are available:

- `ob_start()`: activates output
- `ob_end_flush()`: sends the contents of the output buffer and deactivates buffering
- `ob_end_clean()`: empties the output buffer and deactivates buffering
- `ob_get_contents()`: returns the output buffer contents

These functionalities make it possible to prepare the contents that are displayed via complex processes requiring several database queries, for example, and the sending of results only when all of the queries have been carried out successfully. In the opposite case, you can empty the output buffer contents and display an error message in its place. The example below shows a typical case of how these functionalities are used:

```
<?php
// activate output buffers
ob_start();
// the output is not displayed,
it enters the buffer print
"Hello, world!";
// read buffer contents
$output = ob_get_contents();
// deactivate the buffer, empty
the buffer
ob_end_clean();
// display contents
print $output;
?>
```

Evaluate for Identical Operator

A new evaluate for identical operator makes its debut in PHP4. This operator makes it possible to test the equality of the values and types of two different variables or expressions. With implicit PHP-type conversions, the value 5 could be compared with the string containing the character "5." These two values were considered equal in a comparison. The new operator is represented by the three equal signs (`"= = ="`) and only sends back TRUE if the variable type and value are identical. Thus, in the

(Continued on page 10)

What's New in PHP4? (cont.)

(Continued from page 9)

following example, the four tests using the equality operator with the numbers 1 to 4 are true, while with the following four, which use the evaluate for identical operator, only tests 5 and 8 are true.

```
$valint1 = 1;
$valint2 = 2;
$valint3 = 3;

$valfloat1 = 1.0;
$valfloat2 = 2.0;
$valfloat3 = 3.0;

$valstr3 = '3';

$res = (($valint1+$valint2)==
$valint3) ? ('TRUE') :
('FALSE') ;
print " TEST 1 : " . $res . "<br>
\n";

$res = ($valint3==$valfloat3) ?
('TRUE') : ('FALSE') ;
print " TEST 2 : " . $res . "<br>
\n";

$res = ($valstr3==$valint3) ?
('TRUE') : ('FALSE') ;
print " TEST 3 : " . $res . "<br>
\n";
print " TEST 4 : " . $res . "<br>
\n<br>";

$res = (($valint1+$valint2)===
$valint3) ? ('TRUE') :
('FALSE') ;
print " TEST 5 : " . $res . "<br>
\n";

$res = ($valint3=== $valfloat3) ?
('TRUE') : ('FALSE') ;
print " TEST 6 : " . $res . "<br>
\n";

$res = ($valstr3=== $valint3) ?
('TRUE') : ('FALSE') ;
print " TEST 7 : " . $res . "<br>
\n";

$res = (($valfloat1+$valfloat2)
=== $valfloat3) ? ('TRUE') :
('FALSE') ;
print " TEST 8 : " . $res . "<br>
\n";
```

In passing, it is worth noting that PHP4 now includes Boolean-type data.

COM support on Windows

On Windows platforms, PHP4 now offers support for COM components. This means that practically all Windows applications can be manipulated from PHP4. The COM model that is at the heart of Microsoft's architecture is such that any Windows application can provide a certain number of components and exposed methods, which are entry points for manipulating applications from other applications.

Since a few lines of code tend to be more useful than a long description, let's look at the following example. The code below does nothing less than boot Word, create a new document from it, add text to this document, and then save everything on the disk before quitting.

```
<?php
    $word = new COM("word.
application") or
        die("Impossible to instanti-
ate WordApp");
    print "Word is running, version
{$word->Version}\n<br>";

    $word->Visible=1;

    $word->Documents->Add();
    $word->Selection->TypeText
("This is a test...");
    $word->Documents[1]->SaveAs
("test_com_php.doc");
    $word->Quit();
?>
```

Displaying portions of raw HTML code
Another newcomer to PHP4 is a new syntax that is modeled after Perl's "here printing." It is possible to display complete portions of HTML code by flanking them with either a print or echo instruction and an end label. Unlike Perl, the operator used is not a double less than sign (<<), but a triple less than sign (<<<) so as to distinguish it from the binary shift operator. For example:

(Continued on page 11)

What's New in PHP4? (cont.)

(Continued from page 10)

```
<?php

$version = phpversion();
$valstr = "Example of Here-
Printing\n";

echo "<hr><br>\n";

print <<< mylabel

<h4>PHP4's New Features</h4>
<ul>
<li>ISAPI support
<li>COM support on Win32
<li>Java and Servlets support
</ul>

<p> $valstr <br>Tested on PHP
version $version</p>

mylabel;

echo "<hr><br>\n";

?>
```

Java support

Java support represents without a doubt one of the new features that will open PHP up to existing enterprise systems and applications. With PHP4, you have the possibility of instantiating and using Java servlets and/or classes.

This support is not native to PHP4; it necessitates the addition and configuration of an extension. In order to have this extension at your disposition on a Linux system, it is necessary to recompile PHP4 by specifying the adequate configuration options. These extensions are available in the form of objects that have been precompiled for the Windows environment. These extensions can be downloaded from the official PHP Web site. You must choose the extension that is adapted to the Java version that you use (JDK 1.1.8, 1.2 or 1.3). In our example, we use a simple JDK 1.1.8

In the `php.ini` configuration file, you must add two lines in a new "Java" section as in the following example:

```
[ java ]
java.class.path="c:\jdk1.1.8
\lib\classes.zip;C:\Program
Files\Apache\cgi-bin\php_java.
jar;C:\Program Files\Apache
\htdocs\phpjava"
extension=php_java.dll
```

You can then make sure that the Java support is configured correctly by checking the contents of `phpinfo()`. You should find a section similar to the following:

Directive	Local Value
java.class.path	c:\jdk1.1.8\lib\classes.zip;C:\Program Files\Apache\cgi-bin\php_java.jar;C:\Program Files\Apache\htdocs\phpjava
java.library	java.dll

Then, place the Java classes that you would like to use in one of the directories given in the list of paths specified by the `java.class.path` configuration directive, for example in `C:\ProgramFiles\Apache\htdocs\phpjava`.

Imagine that you have a Java class named `Point`, which represents a point defined by its coordinates and the various required methods (a constructor, default constructor, classic access methods to set and get properties, and a `toString` method). The `Point` class could look something like the following:

```
public class Point {

    // attributes
    public int abscissa;
    public int ordinate;

    public int type;

    // default constructor
    public Point() {
        setCoordinates(0,0);
        this.type = 1 ;
    }
    // constructor
    public Point(int x,int y) {
        setCoordinates(x,y);
        this.type = 1 ;
    }
}
```

(Continued on page 12)

What's New in PHP4? (cont.)

(Continued from page 11)

```

        // Overloaded object method
        public String toString(){
            // Call the method to
display
            return display();
        }

        // Point class methods
        public void setType(int
mytype){
            // Call the method to
display
            this.type = mytype;
        }
        public int getAbscissa()
{ return abscissa; }
        public int getCoordinates ()
{ return ordinate; }
        public void setCoordinates
(int abscissa, int ordinate) {
            this.abscissa = abscissa;
            this.ordinate = ordinate;
        }
        public sayHello()String {
            return "I am a point :-
");
        }

        public display()String {
            return
            "("+abscissa+", "+ordinate+" Type
            "+type+"");
        }
    }

```

Thus, as long as Java support is configured correctly and the bytecode file `Point.class` is in the appropriate place, we can execute the following PHP code:

```

<?php

    $myPoint = new Java("Point");
    echo "<br>Point created at
(0,0) ...";

    echo $myPoint->setCoordinates
(15,15);
    echo "<br>Point moved to
(15,15)";
    echo "<br>".$myPoint->display
();
    echo "<br>".$myPoint-
>sayHello();
    echo "<hr>\n";

?>

```

Which triggers the following display:

```

Point created at (0,0) ...
Point moved to (15,15)
(15,15 Type 1)
I am a Point :-)

```

Even more interesting is that a `PointBis` class can be derived from a `Point` class by overloading its constructors.

```

public class PointBis extends
Point {

    // constructors
    public PointBis() {
        setCoordinates(0,0);

        super.setType(2);
    }
    public PointBis(int x,int y)
{
        setCoordinates(x,y);

        super.setType(2);
    }

    // methods
    public String sayHello() {
        return "I am a PointBis ;
^)";
    }
}

```

In the same way, we can instantiate and use an object of the `PointBis` class as in the following example:

```

$myPointBis = new Java
("PointBis");
    echo "<br>Point created at
(0,0) ...";

    echo $myPointBis-
>setCoordinates(17,21);
    echo "<br>Point moved to
(17,21)";
    echo "<br>".$myPointBis-
>display ();
    echo "<br>".$myPointBis-
>sayHello();

```

Which gives the following results:

(Continued on page 13)

What's New in PHP4? (cont.)

(Continued from page 12)

```
Point created at (0,0) ...
Point moved to (17,21)
(17,21 Type 2)
I am a PointBis ;^)
```

But you might also want to transfer objects to methods of other objects. Let's take the example of a class that represents a vector. Our vector is defined by its origin (a point), its standard and its direction. In Java, the Vector class can be defined as shown in the following example:

```
public class Vector {

    // attributes
    private origin Point;
    private int    standard;
    private int    direction;

    // constructors
    public Vector() {
        Point p1 = new Point();
        setProps(p1, 0, 0);
    }
    public Vector(Point orig, int
n, int d) {
        setProps(orig, n, d);
    }

    // Overloaded Object method
    public toString() String{
        // Call the method to display
        return display ();
    }
    // Vector class methods
    public Point getOrigin () { re-
turn origin; }
    public int getStandard() { re-
turn standard; }
    public int getDirection() { re-
turn direction; }
    public void setOrigin (Point
origin) {
        this.origin = origin;
    };
    public void setStandard(int
standard) {
        this.standard
=standard;
    };
    public void setDirection(int
direction) {
        this.direction = direction;
```

```
    };
    public void setProps(Point ori-
gin, int standard, int direction)
{
    this.origin = origin;
    this.standard = standard;
    this.direction = direction;
}
    public String display(){
        return "( "+origin.
display()+
        " /
"+standard+" / "+direction+" )";
    }
}
```

In PHP, we can instantiate a vector by using the default constructor via the following code:

```
$myVector = new Java("Vector");
echo "<br>Vector created, de-
fault constructor...";
echo "<br>".$myVector->display
()." <p>\n<br>\n";
```

Which results in the following display:

```
Vector created, default
constructor...
( (0,0 Type 1) / 0 / 0 )
```

But we can also use the Vector class constructor to which we will transfer a PHP object representing a Point object:

```
// creation of a point that
will serve as vector origin
$myPoint = new Java("Point",
11, 65);
echo "<br>Point position...";
echo "<br>".$myPoint->display
()." <p>\n<br>\n";
$myVector = new Java
("Vector");
echo "<br>".$myVector->
display()." <p>\n<br>\n";

print $myVector->setProps
($myPoint, 24, 8);
echo "<br>myVector's origin
is MyPoint";

echo "<br> Vector Posi-
tion...";
echo "<br>".$myVector->display
()." <p>\n<br>\n";
```

(Continued on page 14)

What's New in PHP4? (cont.)

(Continued from page 13)

This code triggers the following display:

```
Point Position...
(11,65 Type 1)

( ( 0,0 Type 1) / 0 / 0 )

myVector's origin is MyPoint
Vector Position...
( (11,65 Type 1) / 24 / 8 )
```

We can see that the vector was originally created by its default constructor: a null vector for which the origin is (0,0) having a null standard and a null angle direction. Then, after calling the setProps() method, we make sure that the vector origin corresponds to the coordinates of our myPoint point (11,65).

This simplistic example is a way of verifying whether Java can be used from PHP, whether they are your own Java codes or existing Java classes. This Java gateway has many applications: using CORBA components via IIOP, distributing processes via IIOP, distributing processes via RMI, coding portions of "sensitive" code of our Java applications so that the source code cannot be unveiled, etc.

License

Finally, it is very important to point out that PHP4 is not distributed under the GNU GPL license as was PHP3, but under a specific

license called the PHP License. In essence, the PHP License is quite similar to the BSD license. It authorizes distribution and modification of PHP within open source or proprietary applications. Moreover, Zend, which is an integral part of PHP4, is distributed under yet another license, the Zend License, which was inspired by Troll Tech's QPL.

This license enables authors to take advantage of all of the contributions and corrections made by the community while reserving the right to make money off of the commercial licenses for the Zend library. However, this does not really concern PHP script programmers and designers.

Final Note

Version 4 of PHP is more than just a simple update. It represents a new implementation that necessitated conception and a total rewrite on the part of Zeev Suraski, Andi Gutmans and all members of the PHP core team. With PHP4, we now have a free solution that is of high technical quality, which will most certainly result in extremely rich professional applications.

Credits

The information presented in this article comes from the official PHP channels: the Zend site (<http://www.zend.com>) and official mailing lists dedicated to PHP3 and PHP4. ■



Upcoming Report: Open Source and Free Software

Find out more about PHP and other free software initiatives in our upcoming report dedicated to this ever-growing movement. **This report provides an in-depth look at the Open Source and Free Software that are changing the face of the computer industry.** A global analysis covering the history, philosophy, development, and distribution of free software. Product profiles, integration hints and more...

Find out more about this report at:

http://www.techmetrix.com/products/prod_report4.shtml

A Close Look Under the Hood of B2B Solutions

By Jean-Christophe Cimetiere - CEO
jcc@techmetrix.com

Introduction

In past issues of TrendMarkers, we have looked at IRM engines (January 2000) and the main players in this growing market (February 2000), which generally addresses the Business-to-Customer (B2C) side of e-business. Apart from a few overlapping aspects, Business-to-Business (B2B) needs and solutions remain considerably different from those of B2B. According to Keenan Vision Inc., a San Francisco-based Internet market research company, in 1999, 70% of all Internet exchange transactions concerned B2B, compared to only 30% for B2C. Moreover, the firm predicts that in 2004, B2B will represent 78% of all transactions. (The full report, "Internet Exchange 2000" is available at www.keenanvision.com).

There are plenty of different approaches within the B2B market. All B2B solution vendors promise to solve your business problems, but you must first decide what your business problems are! The following are some of today's most common business needs:

- E-marketplaces (trading-hub)
- Trading partners (automated interactions with multiple partners)
- Supply chain integration and automated procurement
- Content exchange and syndication (in & out)

You may come upon solutions that are specialized in one or two of the above areas, but in most cases, problems are intertwined and the boundary between one need and another is quite blurry. This is one of the reasons why we have decided to look at the problem from a technical and architectural perspective.

B2B Basics

The B2B market is still extremely young and many vendors are trying to grab up as many market shares, and above all, mind shares as possible. Besides customer de-

mand, EAI^[note1] solutions, which allow for the integration of legacy systems, hence leveraging past investments, were the first to fuel the expansion of the B2B market. Today, XML is the most powerful driving force behind the B2B market; it really is the B2B enabler. Everyone seems to agree that XML is the standard format for exchanging messages and data. Everyone also seems to agree that there is a real need for a standard that describes the way in which XML messages are formed. Document Type Definition (DTD), W3C's XML schemas and Document Content Description are all in the running to become this standard. Within the coming month, the situation should be clarified, and we should have a better idea of which of these contenders is the best suited to fill this role. In addition, Biztalk.org, Rosettanet.org and XML.org have taken the initiative to create XML-DTD repositories. These repositories are organized by sector and hope to define typical exchanged data and processes.

But XML exchange format is just one of the huge number of processes that can be put in place. Extracting data from legacy systems, calling methods from new generations of Web applications, formatting XML messages and requests that need to be sent, processing XML messages and received requests... the list goes on and on; and depending on your industry, your role, and your existing systems, the implementation of these many processes varies tremendously.

From a technical perspective B2B is all about:

- EAI and middleware technologies (message-oriented, ERPs connectors, system gateways, and legacy bridges)
- Distributed or non-distributed transaction management
- J2EE-based and Microsoft DNA application servers
- Automated business processes and rules engines
- XML data conversion/transformation, XML data mapping, XML standard schemas, and XML schemas translation.
- Security management

(Continued on page 16)

A Close Look Under the Hood of B2B Solutions (cont.)

(Continued from page 15)

We are witnessing the emergence of a new generation of solutions called B2B boundary blasters. Actually, "new generation" is not an appropriate term because most of the players have existed for a long time. However, this new paradigm has created a real identified market that leaves room for newcomers.

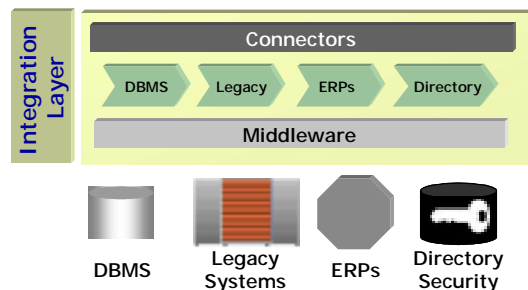
B2B solution architecture

Based on our analysis of available solutions and on our personal vision of this technological trend, we have defined three layers representing the main groups of processes that are addressed in a B2B solution:

- Integration Layer
- Business Process Layer
- Interaction Layer

Let's define each of these layers.

Integration Layer

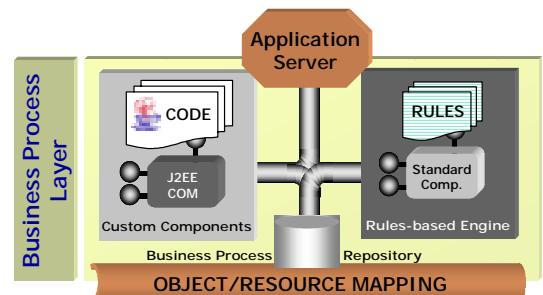


The B2B world is a world of global communication in which you must enable communication from your back-end systems to virtually any other back-end system over the Internet. You cannot master the entire chain of exchanges because you cannot control what's going on beyond your network. You cannot apply the same integration techniques that you use internally to your partners or customers. We'll go into this aspect a little further in the section dedicated to the interaction layer, but it is important to remember that loosely coupled systems are a requirement in the B2B chain.

Before interacting with the world, you must first enable integration and communication between all your in-house systems. This is a truly difficult task that requires many different skills. Indeed, integrating SAP using its Business API is very different from integrating SAP with Cobol CICS. The first aspect that has to be considered is connectivity between systems. Connectivity requires that each system to be integrated have dedicated bricks. However, at the top integration layer, all the connectors plugged into your heterogeneous systems must provide a unified view so as to allow the next step which is building business processes.

Real-time access and interaction with back-end systems is of the utmost importance when it comes to streamlining processes (most store-fronts are still not live-linked to back-end inventory systems). Performance impact and transition integrity are the two main issues that need to be addressed when creating live back-end interaction. Although transaction integrity can be tricky, it is manageable. However, it is much more difficult to address the performance impact of enabling live-access to your back-end system. The main reason behind this difficulty is that it is very hard to predict what kind of load you will have to handle. Caching technologies such as GemStone's Shared Object Cache could be used to address this problem because they limit access to back-end systems. Caching is handled at the next architecture level, the Business Process Layer.

Business Process Layer



This layer is really the heart of the system.

(Continued on page 17)

A Close Look Under the Hood of B2B Solutions (cont.)

(Continued from page 16)

Its first key feature is its object/resource mapping system, which has the role of setting up relationships between integration layer connectors and the objects that will be manipulated in the system. The best mapping method is one that can provide an object-oriented view of all the external resources that are integrated. Sun's J2EE and Microsoft's COM models are the most common options.

Another main feature of the Business Process layer is its ability to quickly define new business rules that will immediately transform into executable and deployable code or components. This is the role of rules-based engines such as Versata products. Rules-based engines provide end users (business people) with tools that are easy-to-use and which enable modification of existing or creation of new business rules.

Custom components handle highly complex business objects. They are developed from scratch using a language (Java, for example) and integrate with the selected object model. This is a vital option within a B2B solution, and resulting components must interact smoothly with rules-based components.

In order to facilitate the use and the customization of the B2B solutions, all business rules, custom components and available resources are referenced in a Business Process repository.

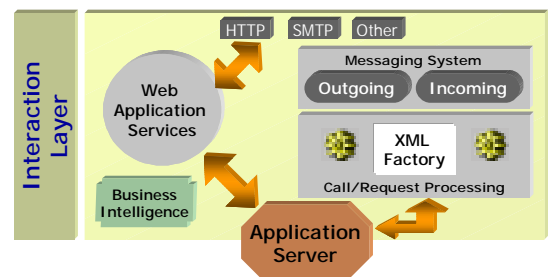
Then of course, all these elements have to work together with the help of the application server. Although the application server is represented as a small entity in the above diagram, it could cover the entire business process layer. It is important to understand that the application server is vital for many reasons:

- Scalability and reliability
- Standard compliance and openness

In the article, "E-business: Reshaping the Application Server Market," (<http://www.techmetrix.com/lab/trendmarkers/tmk0600/tmk0600-1.shtml>) we wrote that

the application server market was shrinking. Actually, it isn't shrinking in volume — the Giga Information Group predicts huge growth in the coming years (see the numbers at http://serverwatch.internet.com/news/2000_06_20_a.html) — but the key leaders of this market are clearly identified, and an application server's added value is no longer enough to make the difference when a customer is looking for the infrastructure on which to develop its e-business strategy.

Interaction Layer



Here again the application server plays an important role. It is the underlying infrastructure of the interaction layer, especially for Web Application Services.

Web application services are the user interfaces your partners, suppliers, and customers use when they need to interact with your company. Typically, these interfaces are traditional Web applications that are handled by the application server and interact with the rules engine and custom components, both of which should be considered the same type of components at runtime. The ability to provide highly customized and flexible applications is essential. The rules engine described in the previous section is also a key element for allowing fast reconfiguration of your services (for example, changing the pricing model for a customer without having to rewrite code).

Furthermore, as mentioned in the integration section, loosely coupled systems are better suited to B2B exchange. They are more fault-tolerant than connected systems because they use asynchronous protocols — and you never know what's going on at the other side. Therefore, protocols that

(Continued on page 18)

A Close Look Under the Hood of B2B Solutions (cont.)

(Continued from page 17)

support the exchanges are mostly Internet-based: HTTP and SMTP. IIOP or DCOM protocols cannot be used for B2B exchange (see our past article about integration: <http://www.techmetrix.com/lab/trendmarkers/tmk0300/tmk0300-3.shtml>). Message Oriented Middleware (MOM), like MQSeries, Tibco, MSMQ, is not standard and may be seen as good solution for the integration layer, but not for the interaction layer.

Above transport protocols are XML messages. XML files can be pure data, or act as Remote Procedure Calls (RPC). Some new protocols such as Simple Object Access Protocol (SOAP) are being defined so as to allow the use of XML in a manner similar to RPC. SOAP was submitted to the IETF as an Internet Draft in December 1999. It is a minimal set of conventions for invoking code using XML and HTTP that may allow any application to call object methods without using classical RMI, IIOP or DCOM.

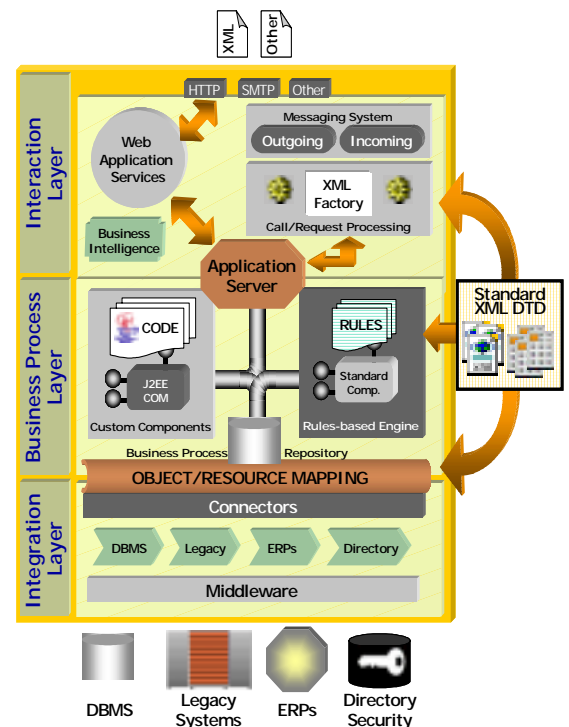
One of the striking features of B2B solutions is what we call the "XML Factory." This service of the B2B engine processes incoming and outgoing XML messages. It can transform XML files into object representations, thus allowing the application server to process according to established business rules or custom components.

The XML factory also easily generates XML messages from business objects, for example. In order to do so, it passes an object representation of a purchase order to an XML file, which can in turn be sent to your supplier's system. The JDOM OSS initiative (<http://www.jdom.org>) is a good example of the basic services provided by the XML factory. This initiative is a Java-based solution for accessing, manipulating, and outputting XML data from Java code.

Business Intelligence units are very useful for auditing and understanding what's happening in your system and for monitoring the interactions with your users. In B2C products, the Business Intelligence unit makes it possible to provide personaliza-

tion services, and suggestions for cross- and up-selling. Indeed, personalization is yet another important aspect of B2B.

The whole engine



Now that we have developed an idea of the whole engine, we need to describe the role XML plays in it. In fact, XML and DTD (or another standard) come into play at all levels.

For object/resource mapping operations, XML can be used as a descriptor by the connectors to facilitate mapping. This is the case with the EJB XML deployment descriptor.

The XML Factory could also execute mapping operations involving two different DTDs representing the same element. For example, your company might use a particular DTD for purchase orders, but your customer's DTD might be slightly different. In this situation, you must establish correspondence between the two DTDs. You can do so by choosing a standard schema available in a DTD repository such as Biztalk.org, Rosettanet.org or XML.org. The system

(Continued on page 19)

A Close Look Under the Hood of B2B Solutions (cont.)

(Continued from page 18)

is still a bit unorganized, but at least it offers some standard schemas.

Rules can also be defined as XML files. A rules-based engine digests XML-defined rules and generates components. It is quite probable that an XML-based standard allowing rules definition will appear in the near future.

In fact, a first initiative was just launched by Intalio, Inc (www.bpml.org) on July 17, 2000. "The BPML.org (for Business Process Modeling Language) Initiative aims at defining a standard way to model, deploy and manage business processes with such Business Process Management Systems," said Ismael Ghalimi, CEO of Intalio.

We strongly encourage corporations to participate in such initiatives.

Conclusion/Market Players

As we have seen, a complete B2B solution addresses many different aspects. As far as we know, no out-of-the-box solution currently addresses every one of these aspects. In the great fight between B2B and

B2C solutions (what we called IRM engines in previous issues of TrendMarkers), the winner will be the one that proves the most adaptable and which has the ability to use third-party tools to make up for the areas in which it lacks.

Throughout this article we have discussed B2B concepts; but what about reality, products and solutions?

The list of vendors that are strongly positioned in the B2B market is long, but we are keeping a close eye on the following: BEA (eLink-WebLogic), Bluestone (Total-e-B2B), BowStreet, Intalio, Excelon, Extricity, Level8, Microsoft (Biztalk Server) NetFish, Software Technologies Corp (STC), Versata, Vitria, and WebMethods/Active Software.

We will start taking a closer look at these products in the next issue of TrendMarkers, so stay tuned. ■

[note1]

Enterprise Application Integration (EAI): EAI covers methods and tools to achieve interoperability, information and processes synchronization across multiple applications: main-frame, ERP, packaged systems, DBMS, and in-house applications, e-commerce storefronts, etc.

Application Servers: GemStone Swallowed Up by Brokat

By Philippe Mougín – Consultant
pmougín@techmetrix.net

It's the end of a nearly 20-year adventure for the company GemStone, editor of the GemStone/J application server and long-standing object specialist. Indeed, Brokat, editor of a renowned e-business platform destined for financial sectors, has just announced it intends to purchase GemStone for approximately \$271 million. This coincides with the company's plans to buy Blaze Software (previously Neuron Data), provider of a rules-based technology, for \$560 million. These two acquisitions should enable Brokat to modernize its tech-

nical infrastructure. Brokat announced that GemStone and Blaze products will initially continue to be produced under their existing brand names.

The history of GemStone Systems is intermingled with that of object technologies. Founded in 1982, the company developed one of the first object-oriented databases. Its strong command of technology and its innovations earned GemStone recognition on which it prided itself throughout its existence—and here lies the importance of this acquisition. GemStone may not be a giant

(Continued on page 20)

Application Servers: GemStone Swallowed Up by Brokat (cont.)

(Continued from page 19)

in terms of sales figures, but it is a major force when it comes to the industrialization of cutting edge object techniques.

Several past and rather well planned strategic turns have made GemStone the company it is today. In the latter part of 1990s, GemStone was faced with the resounding commercial failure of the object-oriented database market. By repositioning its product GemStone/J on the application server market, GemStone was able to survive and even become an important player in this area. This successful shift in position was based on technology: GemStone understood that object-oriented databases were application servers long before the term even existed. In other words, the company was several years ahead of its competitors with its historic product! However, another strategic maneuver helps explain Brokat's acquisition of GemStone. Indeed, in 1996, GemStone realized that Java was capable of becoming the mainstream object language. A few months later, GemStone/J was born. Today this Java application server offers EJB support and implements Sun's J2EE™ specifications.

Despite GemStone's obvious technical qualities, the company did not succeed in capturing a large market share. After having considered an IPO, GemStone finally decided to get in touch with several potential buyers. Contacts with vendors such as Sun did not materialize. In the end, it is Brokat that has just announced its intention to acquire GemStone.

Brokat is the provider of Twister, an application server designed principally for the financial sector thanks to its technologies and application services that are particularly well suited to the security and integration problems characteristic of online financial transactions. The acquisition of GemStone resolves one of Brokat's biggest problems. Indeed, confronted with a highly competitive application server market in which technologies evolve at an astounding pace, Brokat found it lagged behind on several technical infrastructure points. At the same time the arrival of Java, EJB, and

J2EE, brought about customer demands to which Brokat had to respond. Acquisition of GemStone, along with that of Blaze Software, should provide Brokat with one of the most modern Java platforms around and allow the German company to get a more stable foothold in the American market.

Of course, all this depends on successful integration between these different technologies. Certainly, with GemStone, Brokat gains an EJB server, but this alone does not justify an expenditure of \$271 million. Furthermore, this technology is not necessarily GemStone's strong point. We were disappointed by GemStone's EJB implementation during our evaluation of version 3 of the product, which did not keep its promises with regard to this technology. At the time, GemStone R&D was in the process of reworking its EJB technology architecture following some design errors. Another study published earlier this year was just as severe. In particular, this study found that GemStone's EJB implementation did not comply entirely with the specification (the study was based on EJB 1.0 compliance, but for many the results could also apply to specification 1.1). GemStone did not perform as well as the other products evaluated in the benchmark, especially when compared with WebLogic's EJBs. A little aside: After some hesitation, GemStone decided against giving a counter-argument; however, the vendor did share its anger with the publisher of the study, and the presses were nimbly stopped. Having said this, GemStone version 4, which has been available since June, does offer an acceptable EJB server, especially if one uses THOUGHT, Inc.'s object-relational mapping tool, CocoBase, which is specially adapted to GemStone.

Luckily for GemStone customers and for Brokat, the technical value of GemStone/J goes far beyond EJB technology, which in spite of all the hype, is probably down to its last chance of finding a veritable place within the enterprise with the release of version 2.0 of the standard. In fact, GemStone's Persistent Cache Architecture (PCA) is a major technology that surpasses EJB

(Continued on page 21)

Application Servers: GemStone Swallowed Up by Brokat (cont.)

(Continued from page 20)

servers quite considerably. Thanks to its command of object-oriented databases, PCA is virtually unique on the application server market. This technology allows for implementation of the most advanced object technologies, in particular, native object persistence in a transactional environment. When combined with such elements as JSP and security, and with such technology as clustering, PCA offers teams that have split with object technology an application server that makes its competitors look like toys. Brokat has thus just picked up a rare technical pearl. Of course, this technology requires use of a specially modified JVM. Although this might seem inconvenient, it is actually an important advantage. In fact, GemStone has proven in the past that use of this special JVM does not hinder its reactivity in the slightest; but most of all, this shows what a rare command the vendor has of fundamental technical aspects. Generally speaking, industrial command of these technologies remains the privilege of major players such as IBM or Sun. Finally, it is worth noting that PCA is EJB aware, and it is therefore possible to use it in combination with EJB components. This synergy will become more and more interesting as GemStone's EJB technology is improved and will make it stand out from rival servers.

Finally, with GemStone Brokat inherits a collection of products that are destined for the world of Smalltalk. In particular, it picks up GemStone/S, which is a Smalltalk application server that is also based on an ob-

ject-oriented database. Here again, we must bring up the idea of a rare pearl; this product is one of the most advanced existing application servers. GemStone/S offers its users great added value. The past few years have shown that it is one of the few environments enabling the implementation of complex, large-scale object-based projects. This server's power and ease-of-use greatly surpass those of GemStone/J, which as we already mentioned is quite impressive. But GemStone/S is designed for a very particular customer target insofar as Smalltalk is rapidly losing importance in the market. GemStone/S is most certainly not a part of Brokat's strategy, which is focused on other priorities—in particular, trying to make a profit. Nonetheless, Brokat mustn't bury this technology, which will most likely happen if it is relegated to second-class status. The solution is probably to come up with a spin-off or sell it off to an interested vendor.

So, is Brokat's acquisition of GemStone good news? Certainly not for the general-use application server market, which risks losing one of its jewels since Brokat is not really cut out to be a player in this market. However, Brokat has said that it will leave the door open for ISVs to use its new infrastructure on other vertical markets. But this is very good news for Brokat (and its customers) which acquire a technology and unique expertise. Brokat has gambled on GemStone's integration; in order to win big the company must now separate the hype from the veritable technical value and act accordingly. ■



TechMetrix Research
6 New England Executive Park, # 400
Burlington, MA 01803
Phone/Fax: 781-270-7486/87

Assessments and conclusions rendered by TechMetrix Research are proprietary. TechMetrix Research and/or TechMetrix Research analysts cannot be held liable for any damages directly or indirectly caused by decisions made using any TechMetrix Research material.

Names appearing in this document that are registered trademarks are not mentioned as being so, nor is the trademark symbol inserted with each mention of these registered trademarks. This document uses these trademarks for editorial purposes only. In no way does TechMetrix Research have the intention of infringing on any registered trademark mentioned in editorial.

© TechMetrix Research 2000 - www.techmetrix.com