

Realtime
publishers

"Leading the Conversation"

The Definitive Guide[™] To

Business Service Management

sponsored by

COMPUWARE[®]

Greg Shields

Chapter 5: End User Experience Monitoring.....	93
System Counters Alone Cannot Fully Represent the End User’s Experience.....	94
System Counters Are Critical to the Systems Administrators and End User Experience Is Critical to the System Users.....	97
Agent-Based Monitoring	98
Agentless Monitoring.....	99
Understanding the “CNS Spread”	100
Watching How Users Interact with the System	101
An Example	102
Visibility	102
Prioritization	102
Resolution	103
Improvement	103
Impacted Technologies	104
Web Front End.....	105
Packaged Applications.....	106
Thin Client	106
Middleware	107
Databases	107
Importance to IT Goals	108
Problem Identification	108
Prioritization	109
Pre-Failure Warnings	109
“Finger Pointing” Prevention.....	110
Clear Problem Communication.....	111
Vendor Accountability.....	111
Customer Satisfaction	112
EUE Ties into BSM	112
Necessary for a Complete Picture of BSM.....	113
Importance of Using Both Methods for Monitoring.....	114
Proactive Awareness.....	115
EUE Drives BSM’s ROI.....	115

Copyright Statement

© 2007 Realtimepublishers.com, Inc. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtimepublishers.com, Inc. (the “Materials”) and this site and any such Materials are protected by international copyright and trademark laws.

THE MATERIALS ARE PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. The Materials are subject to change without notice and do not represent a commitment on the part of Realtimepublishers.com, Inc or its web site sponsors. In no event shall Realtimepublishers.com, Inc. or its web site sponsors be held liable for technical or editorial errors or omissions contained in the Materials, including without limitation, for any direct, indirect, incidental, special, exemplary or consequential damages whatsoever resulting from the use of any information contained in the Materials.

The Materials (including but not limited to the text, images, audio, and/or video) may not be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way, in whole or in part, except that one copy may be downloaded for your personal, non-commercial use on a single computer. In connection with such use, you may not modify or obscure any copyright or other proprietary notice.

The Materials may contain trademarks, services marks and logos that are the property of third parties. You are not permitted to use these trademarks, services marks or logos without prior written consent of such third parties.

Realtimepublishers.com and the Realtimepublishers logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners.

If you have any questions about these terms, or if you would like information about licensing materials from Realtimepublishers.com, please contact us via e-mail at info@realtimepublishers.com.

[**Editor's Note:** This eBook was downloaded from Realtime Nexus—The Digital Library. All leading technology guides from Realtimepublishers can be found at <http://nexus.realtimepublishers.com>.]

Chapter 5: End User Experience Monitoring

Its 4:05pm, late on a sunny Friday afternoon, and Dan finds himself staring longingly out the window of his office. He's thinking about his upcoming weekend golf game as he finishes up what he believes to be an easy tag-up call with his biggest customer. The conversation starts off well, both executives chatting about recent vacations, swapping golf handicaps, and discussing the health of their respective companies.

But then his fellow executive, Joe Gear of Glass Emporium, lobs a grenade right into the middle of Dan's impending weekend, "Oh, and one last thing. You know, Dan, we've always been frank with each other. We've known each other for years. Heck, GE's bought millions of dollars of inventory from FCG."

"Uh huh..." mutters Dan as he waits for the other shoe to drop.

"I really hate to have to say this," says Joe, "But my people have been complaining more and more to me about the quality of that Web site you're running over there. When it works, it works great. But sometimes it just seems to freeze on us. Since you moved most of the purchasing over to this online store, our people are kind of trapped when your site freezes or we get error messages."

Dan groans silently to himself as Joe continues, "Dan, they're starting to complain to me about it. If the news is getting up to my level that the site's got problems, then you and I both know that your site's got problems."

Stunned, Dan sits back in his chair. He and John have been working hard on this monitoring and alerting project for just this reason. Six months ago his pager was going off all the time, waking him at all hours of the night and irritating his wife to no end. He now sports this new computer monitor on his desk that shows him the status of his systems. It notifies him when areas of the system go down, so John doesn't have to report to him when problems occur.

What's odd is that other than the occasional blip, that monitor hasn't shown anything major in weeks.

"Tell me more about these problems you're seeing," Dan asks concernedly, "You guys are our biggest customer. If your people are waiting around for our Web site to catch up, then that just begs the question of how many of our other customers are just... going... elsewhere..."

Joe hears his pain, "Well, you know I'm no computer expert, but from the reports I'm getting from my people down in purchasing, the site will work just fine for an hour, or a day, or even a week. But then sometimes it'll just freeze. People will be entering in orders and the site will take 3 or 4 minutes to refresh a page. Sometimes instead of an 'order complete' page we just get an error page. More often than not, the transaction is still there when we come back into the interface, and that's good. But all this extra time spent waiting for your system is costing me money with my guys sitting on their hands.

"Now, none of this is intended to be any kind of threat. We're friends. Our business relationship has been great for many years. But Dan, you've got to do something about that Web system. My people are pulling their hair out. Check it out. You should experience it for yourself some time."

Dan thanks Joe for the candid information. They agree to get together soon for another round of eighteen and an answer to this problem. Dan then picks up the phone and calls John for a report from his IT Manager.

“Nothing to report here, sir,” John reports, “Since we updated our monitoring system to watch for performance metrics on the servers, we found a few servers that needed extra memory or another processor. Those were all upgraded months ago. Since then, other than the occasional processor spike, we haven’t noticed much in the way of problems. You should be seeing this too, now that we’ve got that monitor on your desk. You’re seeing the same info that I’m seeing.”

Dan responds, “But I’m getting reports from our customers, just one today in fact, who say that their experience with using the Web site has been really bad. Freezes. Lockups. Error pages. The whole experience isn’t good sometimes.”

“Well, there’s nothing that I can report from this side. We’re running one of the best monitoring platforms you can buy,” John proudly exclaims, “We’re monitoring dozens of performance counters on everything from routers to switches to the servers themselves, and I can’t report on anything that’s out of the ordinary.”

Dan finds himself a bit ruffled by John’s flippant response. He’s concerned about the experience his customers are feeling when they interface with his company. He’s purchased some very expensive systems monitoring equipment. His “other” monitor shows a happy, healthy system. But somehow all of that monitoring equipment still isn’t capturing the essence of his customer’s experience.

“Meet me in my office now.” Dan instructs John, “We’ve got more strategizing to do.”

System Counters Alone Cannot Fully Represent the End User’s Experience

One of the key tenets of BSM is to provide to the business an understanding of its systems and how those systems relate to profitability and the satisfaction of its customers. BSM does this through the concept of Service Quality. In Chapter 1, we talked about how the quality of a service is impacted not only by that service but also its supporting services:

A loss in a sub-system to a business service feeds into the total quality of that service. A reduction in the performance of a system reduces its quality. And, most importantly, an increase in response time for a customer-facing system reduces its service quality.

We talk so heavily about this measurement of service quality because it is that quantitative measurement that helps the business come to understand its customers’ experience with the services it provides. If those services are not of high quality, customers may take their business elsewhere. If services do not perform appropriately, a business’ public relations can suffer. Ultimately, if a business cannot service its customers to the level that they require, that business cannot continue to operate.

One problem, however, with quantitatively measuring that level of quality lies within the device-centric nature of IT itself. In Chapter 2, we talked at length about the maturity of the IT organization and how a maturing IT organization finds itself better aligned with the needs of its business. As we’ll find out in this chapter, another benefit of a maturing IT organization is that they have a much better understanding of the types of metrics that are critical to correctly measuring their services’ quality.

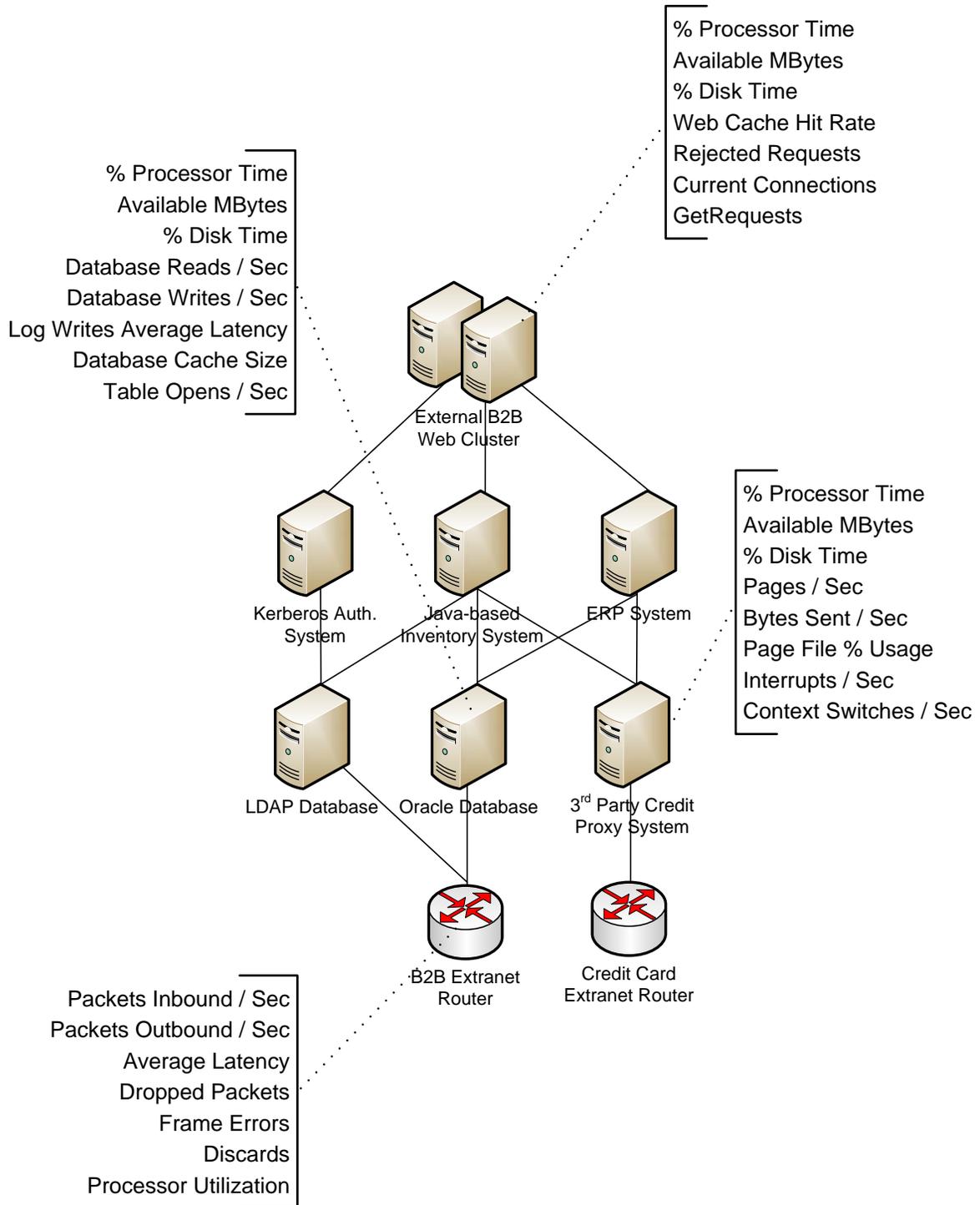


Figure 5.1: FCG's monitoring system is watching for system counters at multiple levels, but those counters aren't telling the story of their user's experience. This figure highlights typical counters often enabled on many systems. But these counters alone don't show the entire picture of what their customers are seeing.

Looking at the Wrong Set of Data

Now what do we mean when we talk about *the types of metrics that are critical*? Take a look at Figure 5.1 and think about the problems seen in our chapter example. In the situation described earlier, John and Dan have worked hand-in-hand to set up system performance counters on their systems. Those counters are watching the servers, network devices, and interconnections to see where performance goes below set thresholds. But, as we see from the phone call with Joe, measuring those counters at the systems level is not necessarily telling the true tale of the users who log in and make use of the external Web system.

In our chapter example, the performance for individual processor and memory use may show that individual systems are performing as normal. We may have plenty of available memory. Our processors may never spike to more than 80% utilization. Our network switches and routers may show a mere 10% utilization of the underlying network. But for some reason, the Web site still slows for users from time to time.

The “Egg Timer” Problem

Obviously something is missing from the equation. What we have in the situation described earlier is much like an “egg timer.” An egg timer performs a very useful task when cooking. This little tool dings when a preset number of minutes has elapsed. That noise notifies us when our eggs are ready to be taken out of the boiling water.

But does that timer actually tell us about the quality of those eggs when the proscribed time arrives? Is the tool looking into the egg to verify that the contents are truly hard-boiled or is it merely telling us that the expected amount of time that an egg should be done has elapsed?

Both of these questions relate to the problems with our chapter example. FCG has implemented a comprehensive monitoring platform that has the rich ability to peer into multiple classes of devices and report on their performance data. They can return and report on metrics that show the health of individual systems and network devices. They can alert when that performance goes above preconfigured thresholds. But are all those metrics—like the ones that Figure 5.1 shows—actually representing the experience of the users on the system? The intent of this chapter is to show that they are not.

System Counters Are Critical to the Systems Administrators and End User Experience Is Critical to the System Users

First and foremost, it is not the intent of this chapter to argue that these metrics are unnecessary. In fact, quite the opposite is true. These metrics are critical for the best possible operations of the servers that run the computing environment. Systems administrators everywhere will tell you that being alerted when a processor is spiked at 99% utilization is necessary to quickly resolve that problem. In the same vein, administrators need to know when a server is running out of memory. Maybe a runaway process is consuming more than its fair share of memory and needs reconfiguring or patching. Maybe the system itself needs additional memory installed to support its workload.

These metrics are critical to the administrators of the system. However, what is also necessary is an additional set of metrics that can represent the experience of a user on the system. Those metrics can be gathered typically through one of two ways:

- First, it is necessary for some automated mechanism to simulate the experience of completing key business transactions. This automated instrument is referred to as an agent, and can be installed onto one or more systems to complete its task.
- Second, it is also critical to get a “big picture” of the entire environment. In order to do so, a tool can be installed into that environment that watches for all the traffic that passes by in that environment. This tool watches for situations in which contentions for resources may be causing problems. Or, it may look for individual transactions that don’t complete or take extra time to complete. It may also recognize when external—and otherwise unmonitored—forces may be contributing to the problem.

In either of these two cases, the concept of a *transaction* is critical to recognizing what this sort of system is looking for. This end user experience monitoring tool needs to look for business transactions or completed interactions between business systems, and how those interactions are behaving. If transactions aren’t behaving properly, there will likely be an impact to the overall operations of those systems on the network. Those delayed or failed transactions may not necessarily impact the overall performance of the server, but they do manifest into the user’s experience.

In Chapter 3, we talked about some of the different mechanisms by which monitoring data can be obtained. Over the years, these different types of data-gathering mechanisms have evolved to provide ever better quality of information through different vectors. Each different mechanism of collection provides data that the others cannot.

For example, an agentless solution can more easily monitor the interrelation between systems over the network than an agent-based solution can. However, due to their installation onto an individual server, agent-based solutions typically have more access to the inner workings of a system. Agent-based solutions can also repetitively execute synthetic transactions to a system to judge their overall performance over time.

Let’s take a look now at these two types of End User Experience (EUE) monitoring classes and how they work. Each can work with the other to get a holistic picture of the system along with its interrelation with the rest of the computing environment.

Agent-Based Monitoring

The goal of agent-based monitoring is two-fold. First, by installing agents onto individual servers that make up a business service, the agents can look deeply into the processes and activities that make up that service. The agent can analyze behaviors within the server to look for individual transactions, the success or failure of those transactions, and the quantity of time elapsed to complete those transactions. Because the agent is installed directly to the specific server of interest, that agent can be configured with relatively unrestricted access to gather and report on the information it needs from within the server.

 This is a very important point—agentless monitoring mechanisms can only query a server through APIs that are published and enabled for external interfacing. These externally facing interfaces do not typically expose all the data within a server, usually for security or functionality reasons. Thus, the addition of agent-based monitoring improves the overall level of information to be processed by the EUE system.

Second, agents can also be installed onto clients throughout the network. The agents on these clients are then programmed to emulate an end-user performing key business transactions throughout the day. Depending on the maturity of the EUE system in place, those instructions may be capable of

- Logging onto a Web site, completing a transaction, and logging off.
- Interfacing with a third-party packaged application such as SAP, Siebel, or other shrink-wrapped software to complete a common task.
- Working with a thin-client application such as Microsoft Terminal Services or Citrix Presentation Server to identify the quality of the server-to-client experience. These same agents can at the same time compare the server-to-back-end server traffic alongside the client-to-thin client server traffic to look for anomalies.
- Synthetically manipulating records within a database or through a middleware software package to identify areas of performance lag.

By installing these agents on systems across the network, the EUE system can compare the results of each transaction with those of other agents to see where individual sites may be experiencing problems. In many ways, the idea with agent-based solutions is to determine the total time necessary to complete a transaction from multiple locations to help identify the characteristics and locations where poor application performance is experience.

 Agentless monitoring, which we'll discuss in the next section, can require very little setup time to configure. However, as you can see with agent-based monitoring, there is a period of configuration necessary to identify the transactions of interest and record them into the agent. For mature EUE software packages, this recording process can be relatively easy. The hard part is in identifying the applications and transactions that are of monitoring interest to the business.

In Chapter 4, we discussed the seven step process to implement a BSM solution. Many of the same processes that are used to build a BSM service model can be leveraged to assist in the process of identifying the right transactions and service components to monitor. As with the service model creation process, this transaction identification activity will be an organic, iterative process.

Agentless Monitoring

Much different than agent-based monitoring is the concept of agentless monitoring. Here, code is not installed to the individual servers that make up the business service nor are any transactions synthetically generated to the systems under management. Instead, we leverage a central solution that is configured to watch for all the traffic across the network. Once installed, the service begins to look for a series of known metrics that can occur across the network:

- When did a particular transaction start? When did it stop? Between what two servers, services, and applications did it occur?
- Did the transaction complete?
- If the transaction did not complete, was it because the user cancelled it or was it due to a network problem or poor performance?
- If the transaction did complete, did it do so within an acceptable amount of time? How much time was spent on the server, the network, and the desktop?
- What are the network conditions across all hosts? Is one host consuming inordinately more bandwidth than normal? Why is that occurring? Is that consumption affecting transaction completion?



A concern in some networks is the promiscuous nature of agentless monitoring. An agentless EUE tool is indeed watching for many (and sometimes all) traffic types in a particular network segment. In some environments, this may go against established security policies. Thus, there may be political pressure not to incorporate an agentless tool due to the type of collection it is performing. That being said, the benefits associated with an agentless monitoring solution must be placed against the security liability associated with allowing it on the network. In addition, although the monitoring is promiscuous, many agentless monitoring tools operate by inspecting just what they need in the network traffic and retaining only the information necessary to classify the results of that inspection. In addition, the agentless monitoring tool should have the ability to mask out sensitive information such as passwords. In many cases, the benefits to the organization far outweigh any perceived security risks.

This agentless solution, in combination with the business logic programmed into the BSM service model, will determine the business impact associated with any transactions that did not complete properly or within a proper amount of time. When a transaction does not complete properly or timely, the service's quality is reduced. Wrapping this idea into the greater picture of BSM, the reduction in service quality directly impacts the dollars-and-cents calculations provided by the BSM system.



We'll talk more about the interconnections between EUE and performance logic and BSM's financial logic later in this chapter.

Obviously, in order for this system to do its job, it has to understand the traffic it is receiving. If a Web server is communicating with a Web browser client, that traffic needs to be understood as a Web request followed by a Web server response. It can also be a series of requests and responses that make up a complete business transaction. This type of communication is programmatically easy to understand. Where mature EUE systems provide extra added value is when those systems can additionally translate non-Web application traffic.

For example, if the EUE system understands the communication that occurs between an SAP system and an Oracle back-end database, it can watch the traffic between those two systems and look for individual transactions. The same holds true with any packaged application. When considering an EUE system, consider one that includes the “special decodes” that can translate traffic as necessary between the systems that ultimately make up the business’ service model.

 As you can probably guess, for a system that is watching traffic all across the network, the sheer mass of traffic that system needs to process is huge. One of the most critical parts of an agentless EUE system is merely to know what kinds of traffic to process and which to discard.

Understanding the “CNS Spread”

In the end, the most important part of this system is in converting this huge quantity of data into something that is useable by its users. In many cases, individual issues within a business system can be related to one of three potential locations:

- The clients that are attempting to make use of the system.
- The network that allows those systems to communicate with each other.
- The servers that process and respond to requests

For any issue that is raised by an EUE system, the problem most often can be related to one of these three elements. As an example, for a transaction to complete, there is a quantity of time required for the client to make a request, the network to transmit the request, the server to receive and respond to the request, the network to return the response, and the client to process that response. A correctly implemented EUE system should be able to provide a “spread” of timing information for each of these elements.

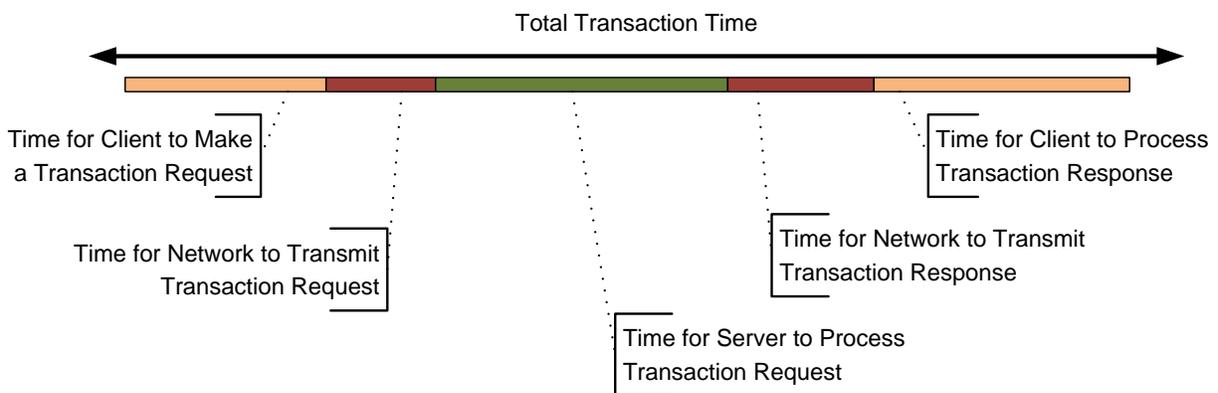


Figure 5.2: The total time necessary to complete a transaction is comprised of multiple steps in the process. The “CNS Spread” identifies each of these elements and their relation to the total transaction time.

This information on the “spread” can be used in multiple ways. As a troubleshooting tool, it comes in handy for isolating where the problems with transaction processing are occurring. As a component of a notification system, it can alert administrators when individual components of transactions are not completing within specifications. As a Help desk mechanism, it can be used to assist users with identifying why their experience is not at their normal level. Most important, this information can be used as a first step in understanding the true nature of the users’ experience and what elements are driving that experience level.

 Figure 5.2 shows only a very simple example of the spread. This example shows the interaction between a single client and a single server. Most business systems and their transactions involve the communication between multiple entities to complete a transaction. It is that interrelation that can be captured by EUE monitoring and is one of its greatest value propositions.

Watching How Users Interact with the System

Even more important is that EUE need not necessarily be strictly a tool for troubleshooting and remediation. Once its elements are set into place, an additional level of visibility is gained into how the users of the system are interacting with the system. Think of the situation we’ve set up already in this chapter. An EUE system with both agentless and agent-based tools is monitoring the user’s experience and the overall network health from the perspective of individual transaction. That system is reporting on the health of each transaction as well as its status.

This information can also be gleaned for useful metrics data as well, telling the business how customers are making use of their on-line systems. For example, maybe customers tend to steer clear of certain areas of the system. This may be due to a design decision or a challenge in the user’s workflow that they naturally do not use. Maybe the users aren’t interested in particular system areas.

A fully-realized EUE implementation can help the business in learning which areas of the system are interesting to users. The business can then strategically recognize and exploit those interests for additional financial gain or better customer satisfaction. Conversely, the same system can look for areas when users bail out of the system. At how many seconds of delay does a user go elsewhere? When a particular page or screen is sent to the user, does that user interact with the supplied page or screen, do they navigate away from it, or leave the system entirely. This kind of user-specific information can be critical towards improving the user’s experience with the system, and ultimately the bottom line of the business.

 Thus, an EUE system can be as useful for the marketing department as it can be for the IT department and business leadership.

An Example

So implementing EUE doesn't necessarily replace typical system counters used by IT in measuring the total performance of their systems. Instead, it adds a new class of counters that watch for individual user interactions with the system. As users interact with the system, an EUE system can measure those interactions—on a click-by-click basis—to ascertain a feeling for what the overall user's experience is with the system. Though much of this measurement is involved with the measurement of elapsed time and time delay, this is not the only tool.

Time tells the tale of how much time users are waiting on system elements, but the experience also relates to individual transactions that don't complete or only partially complete. The true tale of the user's experience is the aggregation of all these metrics.

Let's take a look now at what might have occurred had FCG implemented an EUE system to augment what IT Director John called "the best monitoring platform money can buy."

Visibility

With traditional systems monitoring tools, the counters being measured are based on the performance of the entire system. So those counters may not necessarily pick up problems when they aren't of a nature large enough to affect the system as a whole. System counters typically watch for resource overuse. But the timing delays that EUE is watching for typically don't result in that level of resource overuse. So, the visibility into the specific type of problem FCG's web site is experiencing is not being measured by their whole-system counters.

Had FCG implemented an EUE system to measure timing delays, their system would have picked up on the individual transaction delays that caused users to wait multiple minutes between clicks. That visibility would have alerted them to look for problems at a lower level of the service model. Perhaps a piece of un-optimized code within the purchasing system was causing a counter to time out in certain circumstances. The delay associated with that counter's timeout could have been at the root of the problem. Only an EUE system can peer deep into the individual transactions to see the precise conversation in which that counter's delay occurred.

Prioritization

Because FCG's system didn't include EUE monitoring, and because the problem didn't impact whole system counters, they were unaware that the problem was even occurring. FCG was unable to prioritize resources towards fixing the problem because of their lack of visibility.

In other examples, EUE monitoring may identify multiple locations in which problems are occurring. But they also provide data as to which systems are truly affecting users. If a dozen open problem tickets are created by the help desk associated with issues on the web site, EUE can help identify which of those problems are actually affecting the user population. This grants IT the ability to assign resources first to the problems with the greatest business impact.

Resolution

IT administrators can't fix a problem when they can't see the problem itself. Lacking the tools that dig deeply into each individual transaction, it is challenging to identify problem root causes. Because whole system counters do not necessarily completely describe the workload being done on a particular network device, it is necessary to use tools that can.

EUE's agent-based tools have the ability to simulate transactions between a representative user and the system itself. Those transactions can be run automatically throughout the day and from multiple locations to form a representative understanding of how a sample user might be experiencing the system. Lacking this capability, administrators would need to regularly and manually click through the system to get a feel for its health.

Specific to each measured transaction is its spread of timing information between ownership by the client, network, and server. This spread is an excellent starting point for locating deviances. Drilling down from that point, additional debugging information specific to the transaction can be viewed by the administrator to further isolate the problem. Deconstructing the problem in this way speeds resolution because it helps to focus troubleshooting efforts to the specific issue at hand.

Improvement

Lastly, once the problem is known, it is easier for IT to identify how best to resolve that problem. IT's typical response for many problem is to add additional hardware to the environment to support added load. But in many cases with complex systems this is not the most effective fix. Where traditional monitoring shows no problem but EUE monitoring shows a delay, the problem may not be attributed to a hardware resource shortfall. It may be attributed to a code fault or a misconfiguration. EUE tools allow IT to more correctly improve the system without defaulting to costly hardware expansion as its only tool for resolution.

Impacted Technologies

Among other elements, the value of an EUE system is directly related to the types of service classes that system can interact with. For example, an EUE system that is limited to web traffic only will lack critical visibility into the packaged applications and legacy programs that typically interact with back-end servers. When an EUE solution cannot translate the communication that occurs on the back-end, then a complete vision into each transaction is not fully recognized.

Let's take a look at five classes of business services that are typically part of a typical business computing environment. For each, we'll analyze how an EUE system can impact their operations.

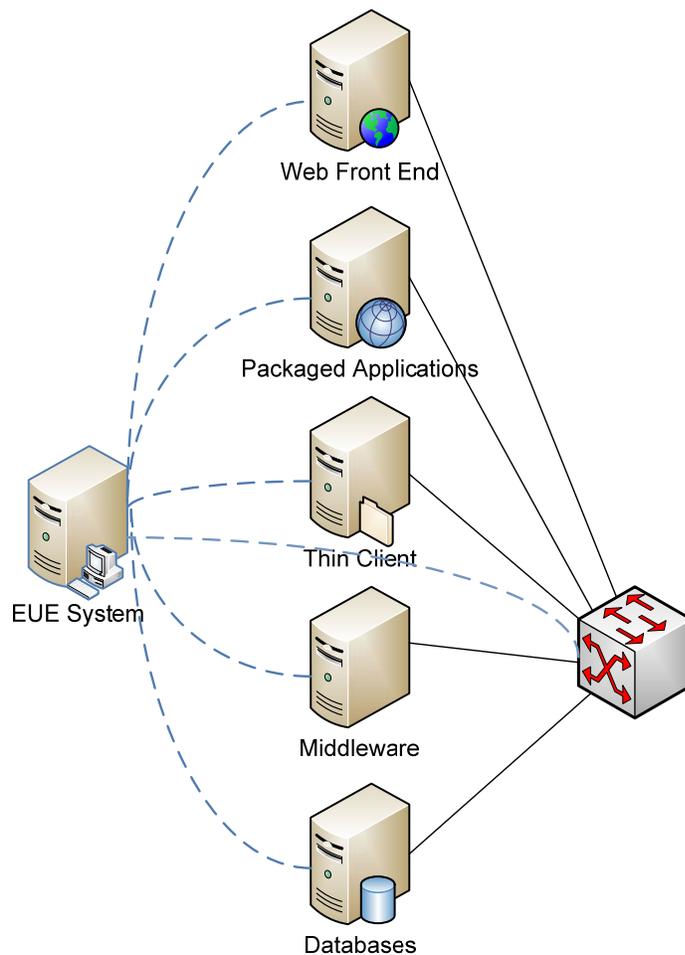


Figure 5.3: A fully-realized EUE system should tie into multiple classes of business services as well as the network they reside upon.

Web Front End

Arguably the most visibly useful for external-facing services, web front ends stand to gain substantial benefits through implementation of an EUE system. The web is nearly exclusively the mechanism for external e-commerce, and thus is the greatest candidate for inclusion in a BSM system due to its impact on business profit and loss.

From a technical perspective, web-based front-ends are also relatively easy to monitor for traffic. This is due to the fact that web traffic is highly transaction-based and easy to translate into a usable form. Due to these paired benefits web front-ends were the target for early implementations of EUE solutions.

With web front-end systems EUE has the ability to “see” into each user’s interaction with the system. Due to the fact that nearly each change of state in a web system involves a click on a web page by a user, the result of these clicks can be monitored and reported on. The interaction between the client, the network, and the web server can be analyzed to look for problem areas or locations where delays are experienced. Those delays can relate to the server not responding quickly enough to a request for more data, or they can be related to a user reading a page or trying to locate the next click in the interface.

This meta-analysis of the entire process of navigating through a web front end can assist the systems administrator with the task of managing the system itself. It can also assist the webmaster with building a web site that is friendly to its users. By analyzing the click-through patterns of the site’s users, the webmaster learns where additional effort or redesign is needed to improve the overall experience.



As web front ends are typically the face of the company for e-commerce, they are typically the low-hanging fruit for an EUE implementation. They also closely tie into the BSM model that EUE feeds its data into.

Chapters 6 and 7 will go into more detail on achieving this management and operational value out of a BSM implementation. EUE and the data it provides is one component of the overall BSM service model.

Packaged Applications

Most business systems don't stop with just the web server. Web front ends typically require additional data from one or more enabling back-end services. In many cases, those services are packaged applications like SAP or Siebel for ERP data, or Oracle plus Oracle Forms for database connectivity and customized business applications. Unlike web services, where all web traffic relies on the common HTTP protocol for data transport and rendering, these packaged applications may have their own protocols for getting data from the client to the server. These applications may not necessarily use a web browser as their data rendering tool at the client side. They may have their own desktop clients that have additional and/or different functionality. Thus, the EUE system used to watch the traffic for these sorts of applications needs to understand the traffic that occurs between client and server.

An effective EUE system will come equipped with the translations or “special decodes” necessary to see into the traffic between the servers and clients that make up these packaged applications. For packaged applications that use multiple servers for distribution of various workloads, the EUE system will also need to understand the server-to-server communication as well. This is necessary because not all issues are directly related to the first-tier client-to-server traffic. Some issues may occur between the individual servers that work together to make up the total service provided by the packaged application.

When considering an EUE system, look for those that can support the packaged applications—typically enterprise-level applications—that are components of your BSM service model. Good EUE solutions should support easy-to-use connectors that allow for the direct listening for traffic between all elements of your packaged applications in relation to clients and any web front ends.



Be careful with some EUE solutions. They may only include monitoring of web transactions. This limitation will restrict the level of information you may require out of your packaged enterprise applications.

Thin Client

One class of packaged applications that requires special attention involves the delivery of applications through a thin client interface. These applications such as Microsoft Terminal Services or Citrix Presentation Server are positioned in front of applications to reduce the overall effects of network latency of bandwidth required to deliver that application to its users.

Consider the situation where the network traffic—the conversation—between an application's server and its client is particularly “chatty”. In this case, positioning the client far away from the server in terms of network proximity means that that application's response time is negatively affected. Because of the network distance between the two halves, the traffic takes a longer amount of time to get from client to server. This increased time means that the client will operate much slower than in the case where the client is close in proximity to the server. Thin client applications relieve this problem by positioning the client next to the server and passing only screen updates and mouse/keyboard movements between client and server.

The use of EUE for thin client applications is multifold. First, in situations where applications are experiencing poor quality, an EUE system's CNS spread can be used to determine where the delays are occurring. If it is determined that the client and server would perform better when they are closely positioned, then EUE can justify the move to a thin client solution for the problem application.

Also useful with thin client applications and the analysis of EUE data is the determination of data problems for existing thin client solutions. Due to the aggregation of multiple users onto a single server in most thin client solutions, the actions of one user can impact the experience of others. For example, one user whose activity on the server uses too many processor resources will cause a slowing down of performance for all others on the server. A fully-realized EUE implementation can be used to determine if the problem relates to the thin client server, the application server, the network between them, or the network between the thin client server and the client itself. In another example where only a single server in a farm is experiencing a problem, EUE can assist with isolating the problem server to help with a quick resolution.

Effective EUE systems should also be able to align the traffic in such a way to isolate user-specific traffic not only from client to server but also thin client server to back-end server. By isolating traffic in this way, an end-to-end understanding of the traffic patterns can be used in troubleshooting and remediation.

Middleware

Although middleware is not always an easy win for an EUE implementation, its incorporation can benefit from many of the same factors associated with packaged applications. As end users do not necessarily work directly with the environment's middleware tools and code frameworks, their incorporation into the total environment analysis can be challenging. However, incorporating middleware monitoring into the overall EUE system ensures that the end-to-end transaction is being monitored. An effective EUE system will include modules that allow connection into the pluggable frameworks that make up most middleware.

Databases

Databases are similarly challenging as are middleware applications. Though they can be a critical component to the overall performance and experience measurement process. As databases contain the whole of the data needed by the business system, their inclusion can be critical in determining the overall health and performance of that system.

Databases that are overloaded in terms of raw performance can specifically impact the delay associated with all other members of the system. This is due to their nature near the bottom of the service model. Inclusion of necessary database monitoring capabilities will help ensure that the full transaction measurement includes client to server to data store, and back if necessary.



In addition to all these, it is also worth stating that the network itself and the devices that make up that network are an impacted technology. Individual network components and their performance can have a net effect on the overall measurement of user experience.

Importance to IT Goals

Thus far in this chapter we've talked about the utility of an EUE implementation and how it relates to the business as a whole. But there are specific benefits to IT that can be gained as well. Traditionally, IT has relied on systems management and monitoring tools to provide them with the necessary information they need to troubleshoot their environments. However, as we discussed earlier with our "egg timer" metaphor, those tools provide shallow levels of data. A mature IT organization will recognize the need for deeper levels of monitoring data to assist with the administration of its systems. That same IT organization will see how the concepts of EUE can provide that data by digging deeper into the individual transactions associated with a business service's operation.

In this section, let's take a look at a few of these benefits specific to IT that can be gained by implementing EUE. From aiding in problem identification and prioritization to augmenting pre-failure warnings, EUE provides a framework for problem isolation. From an organizational standpoint, its information also helps in speeding the troubleshooting process by eliminating the "finger pointing" problem and aiding in inter-team communication. Most importantly, these work together to enhance vendor accountability and ultimately customer satisfaction with the system.

Problem Identification

Traditional monitoring systems have the capability to alert when a problem situation or SLA breach occurs. However, the alert that these systems provide is typically limited to the individual situation that tripped the alarm. Digging deeper into the problem's root cause is limited, because an alerted problem can be comprised of multiple, individual sub-problems, or can be one that is buried within another layer of the system. It is due to these limitations into visualizing the problem that the major time element associated with many problems is simply identifying *what went wrong*.

As we discussed earlier in this chapter, EUE's focus on transactions means that a user's issue with the system can be understood from many different levels. The spread of an application's use of client, network, and server resources is an excellent starting point for the identification of a problem's root cause. This spread provides the troubleshooting administrator a more defined starting point for tracking down the resolution to the problem.

Moreover, digging deeper into each individual transaction allows the system to alert the administrator when problems occur at every step along the path of the transaction. Deconstructing each individual mechanism that makes up the business system helps with the atomization of each service element. This process of deconstruction is very similar to the process used in generating the BSM service model.



Incorporating the necessary thresholds for this alerting is a necessary component for the administrator to complete. Determining what those alerting thresholds should be can be a time-consuming process. However the benefits of knowing when transactions are not within specifications often outweigh the effort.

Prioritization

Even in mature IT environments there are situations where multiple alerts go off at once. When this occurs it can be problematic to understand which of these alerts are important to the functionality of the business and which are of lesser importance. For example, there may be a dozen alerts active within the management system, but eleven of those alerts are actually minor problems that do not require immediate attention. One of those alerts could be one that impacts the entire user base for the business. The process of understanding the true nature of the alert and prioritizing its remediation can be augmented with the information brought forward through an EUE system.

Due to an EUE system's tie into BSM and the BSM service model, each element that makes up the business service under management has an impact assigned to it. Those impacts relate to the number of affected customers and the amount of dollar loss associated with a reduction of service quality. When the situation occurs where multiple alerts are presented, EUE and its tie into BSM helps the IT department understand the business impact of each alert. With this information, IT then has the resources it needs to resolve the most critical and impacting issues first, while de-prioritizing less critical problems.

Pre-Failure Warnings

It is common that a user interface experiences a period of pre-failure before an actual failure occurs. This pre-failure period may relate to an increasing load on the system or a component that trending shows will soon not be able to keep up with the demand placed upon it. What is not common is the recognition of this condition occurring before the failure actually appears. In these situations, only comprehensive trending and historical analysis can assist the IT department with finding these issues before they happen and augmenting the system with additional resources as necessary.

Too often with IT organizations at lower levels of maturity, service failures occur because IT does not have enough information available to recognize when a system requires additional resources, more computing power, or a reconfiguration. EUE can provide that information by continuously monitoring the environment for transaction timing. Trending analysis can be done for service and individual component performance related to transaction speeds. When that analysis points to an impending failure at some point in the future, IT is better prepared to add additional resources as necessary. This also enhances the budgetary process, as fewer surprise purchases are necessary for IT to maintain the environment.

 Consider the following non-IT example as a metaphor for this situation. What if the power company didn't monitor power usage in various parts of the grid? Lacking pre-failure and trending analysis of power usage could mean that building and expansion in certain areas could cause a major loss in its ability to serve power.

As IT organizations mature, their services grow towards a utility status similar to the power company. In these cases it is possible for IT to maintain always-on service, planning for expansion rather than being forced into it by external forces.

It can be further argued that as the IT organization matures, the business matures with it. The business ultimately grows to require this always-on capability as IT discovers the ability to provide it.

“Finger Pointing” Prevention

When critical situations occur with a business system, business revenues are on the line until the problem is fixed. Every second counts in these situations, so solving the problem quickly is critical to operations. The problem within many of these situations, however, is that the typical response by IT is to “get everyone into a room and break down the problem.”

This isn’t necessarily a bad mechanism for isolating a complex problem. IT individuals typically have experience within a single component of IT. “I’m a network person. You’re a server person. Over there are the database people.”

Few individuals truly understand the entire system from end-to-end with the technical know-how to understand problems as they occur. Thus, the “circling-the-wagons” approach in many organizations is the only way to get enough experience in one location to track down the problem.

The problem here is involved with IT personnel ownership of their piece of the computing environment. Professionalism on the part of individuals means that each person in these meetings can default towards proving why the problem does not lie within their scope of management. Individuals in these meetings are incentivized by professional pride to find the problem in other areas of the computing environment. This, combined with the stress of the problem itself, can lead to “finger pointing” within the group, each person trying to find the problem in other areas of the environment.

EUE assists with the “finger pointing problem” first and foremost through the information gleaned through its CNS timing data. Here, when a problem occurs that is critical to operations, the first step can be to look for where the transactions’ client, network, or server times vary from the baseline. The timing information across multiple systems and multiple platforms assists the troubleshooting team with more quickly tracking down the problem.

Even more important is the expensive nature of the group meetings themselves. Considering the per-hour cost of bringing together large numbers of people to identify the problem domain costs the organization in time and money. The opportunity cost of bringing key members of IT together for problem resolution is the effort spent on either fixing the problem or performing other necessary critical work. In organizations with lower levels of maturity, these major problems can occur often. Here, IT finds itself in a state of perpetual “firefighting”, which limits its ability to move towards higher levels of maturity.



A fully-realized EUE system can free these senior-level resources to enable them to work towards strategic, maturing activities rather than tactical, firefighting activities.

Clear Problem Communication

Aligned with the section above, EUE additionally assists with providing a clear distinction between problem domains. IT individuals typically grow deep skills within their scope of management. Areas outside their scope of management have a different vocabulary as well as processes for administration and troubleshooting. Thus, when problems occur that span across multiple scopes of management, the conversation between IT individuals grows complex and adds to the problem. For example:

- The Cisco administrator doesn't speak Windows Server.
- The Windows Server administrator doesn't speak Oracle databases.
- The Oracle DBA doesn't speak SAP.
- The SAP administrator doesn't speak Cisco.

Very few single individuals, especially in enterprise environments, speak the language across all the layers of a business computing environment. Thus, a centralized framework is necessary that can speak some elements of all the necessary IT languages. That framework assists for locating and isolating issues as they appear, but more importantly it is recognized as one that can talk to each individual in their primary IT language. An EUE system is a potential framework that can support this functionality.

Vendor Accountability

Another issue entirely is involved with “holding feet to the fire” for vendors and their applications that the IT organization must support. In most organizations the computing environment is made up of a number of individual applications that work together to provide the business service.

One common problem with this cross-pollination of applications is the tendency of individual vendors to throw an issue “over the wall” when support is requested. As an example, the database vendor suggests that the problem is related to the middleware component. So, a call to the middleware component is necessary. The middleware vendor believes the problem lies within the operating system. So, a call to the operating system vendor support is necessary. Getting all three of these vendors on the phone at the same time—and more importantly the correct people within the vendor's support organization—is challenging if not impossible.

Support technicians associated with many vendors are often incentivized by closing cases rather than fully completing them. Thus, some vendors will tend to throw issues over the wall rather than work them through to completion. This is particularly cumbersome when multiple components of multiple vendors are part of the same business service. It is often functionally impossible when large levels of business-specific customization are done with the vendor's product.

The data provided by an EUE system provides easily-transferable documentation about the behavior of a vendor's application. Data from the EUE system can be provided to the vendor as clear documentation that the problem lays within their product. In some cases, this information can be used to assist with directly pinpointing the problem within their code. When code issues and custom vendor patches are necessary to fix a particular problem, this documented evidence is essential.

This data helps in convincing the vendor that the problem does indeed require a code revision. This same data can also then be used by the vendor in identifying the area in which the fix is necessary.

Customer Satisfaction

Most importantly, all these elements tie into the BSM tenet of customer satisfaction. A service with a high level of quality directly relates to improved customer satisfaction of that service. When IT can proactively identify issues and resolve them without attracting the notice of the user, then they are working at a high level of maturity. That high level of maturity helps IT align better with the needs of the business, and ultimately drive business profitability.

EUE Ties into BSM

It can be argued that for the business leaders of an organization, EUE is all that matters. These individuals care most about the servicing of their customers and how their customers are impacting the bottom line. When customers are not feeling the best possible experience through a business' systems, then it is the business leader who gets the call. At the same time, when customers enjoy working with the business, they tend to keep coming back. It is due to this prioritization that EUE is in many ways an inseparable component of BSM.

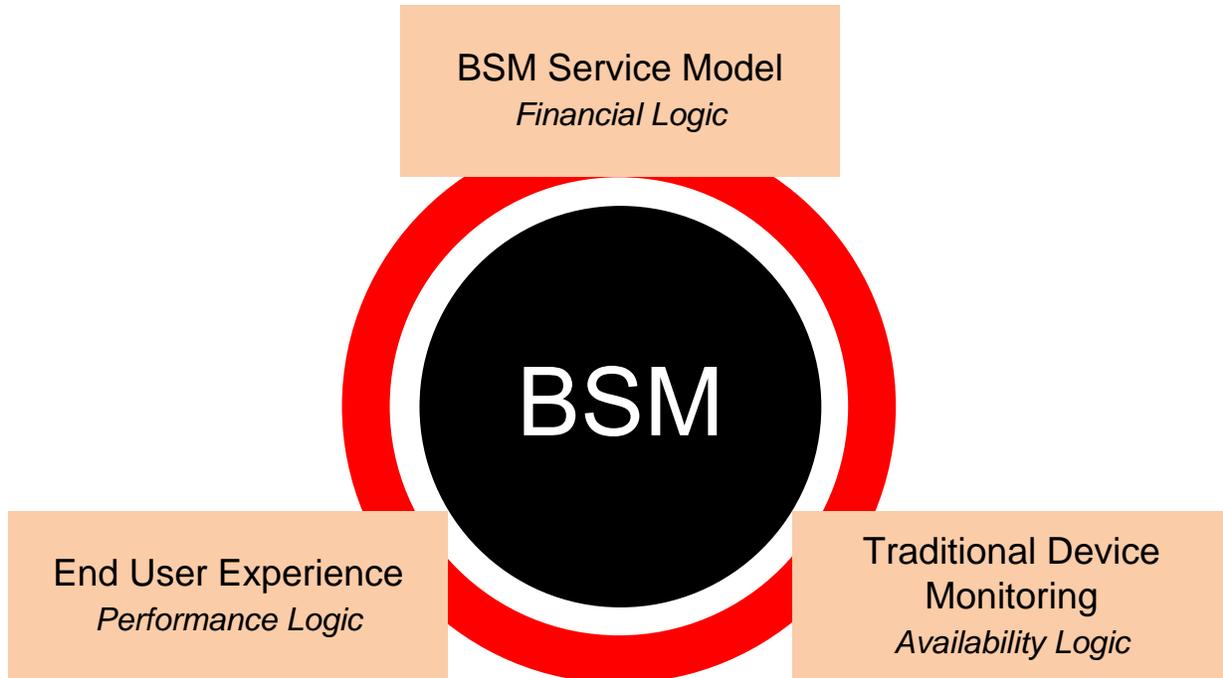


Figure 5.4: Data from three components come together to fill the BSM picture: financial logic from the BSM service model, availability logic from traditional device monitoring, and performance logic associated with end user experience.

Necessary for a Complete Picture of BSM

We've talked in this chapter how the EUE model for monitoring a user's experience draws much of its framework from the same elements used to build the BSM model. This is not done by accident. As we discussed in Chapter 1:

[BSM] ingests availability and performance data and outputs quality-related metrics to the business on the health of the network's business services. BSM applies a dollar value to the reduction in quality of each identified service and serves up the information in the form of dashboards viewable and understandable by both IT and business leaders.

As you can see here, and as pictured in Figure 5.4, the incorporation of EUE into BSM provides a major portion of the data that feeds into the service model. The service model itself is a construct of BSM, and it is that service model that feeds the financial information associated with a business service's quality to the right people. But, there must be a set of driving data that feeds quality information into that model. The combination of the financial logic of the BSM model, individual device monitoring that comes through typical systems monitoring solutions, and the experience state logic associated with EUE is what ultimately fills out the entire solution.

Importance of Using Both Methods for Monitoring

Though not absolutely critical to the overall functionality of the tool, combining the use of both agent-based and agentless methods is what illuminates the best picture of the environment. Agent-based solutions are necessary for driving simulated accesses into the system along with measuring the resulting behavior.

Agent-based solutions are important so that critical timing information specific to user's individual transactions can be measured from multiple points across the network. Also, agent-based monitoring provides a level of on-system data collection capability not possible through external means. For many applications that make up business services, the exposure via external means is simply not available for data collection. Thus, an external source cannot probe for some kinds of essential data like an on system source can.

Additionally, the process of installing agents and their synthetic and simulated transactions across the business network allows for a measure of user experience at multiple points. By depositing agents in multiple locations, the experience specific to that location can be measured in comparison with other locations in the computing environment. These types of measurements assist the CNS spread with isolating issues that may be geographically related.

Agentless monitoring is also crucial to obtaining the entire picture within the environment. Whereas agent-based monitoring can simulate user load and approximate the user's experience through an automated function, it does lack the visibility of the "big picture". An agentless monitoring tool will gather network traffic statistics from all across the network. Its utility is in helping to find when other elements within the network may be the source of the problem.

For example, an agent-based simulation can show that a delay is occurring in a particular transaction. But if that delay is occurring within a system not in the scope of management of the agent, it may not be able to ascertain the root cause of the delay. Agentless monitoring can wade through the network traffic to identify a solution outside the scope of the business service. Lacking agentless monitoring, it is difficult to get this "big picture" view of the entire computing environment.

Proactive Awareness

Lastly, all this data is useless unless the business acts upon it. Knowing that a particular business service is experiencing a loss of quality is only useful when IT knows what to do. Proactive awareness is a function of higher levels of IT maturity due to IT's enhanced knowledge of the components that make up the computing environment:

- IT has the historical data with which to understand how the system and its users evolve over time.
- IT is more prepared from a technical perspective to impact its bottom line from a budgetary perspective.
- IT has more capability for measured and planned growth rather than 11th hour funding requests when emergency resource needs arise.
- IT grows more capable of working with the business on strategic activities like business growth and service expansion rather than merely fulfilling service requests.

All of these relate to IT's ability to better service the business and the customers of the business. By being more proactive with the resources under its care and feeding, IT grows more capable of making better business decisions.

EUE Drives BSM's ROI

As has been shown in this chapter, EUE is a major component of a successful BSM implementation. It provides one very important facet of data needed to make decisions about the efficacy of the business service under management. It illuminates the plight of the service's users, giving the business an automated understanding of how effective they are servicing their customers. And it ultimately provides the necessary financial data to business leaders to help them understand the impact their systems have on the business' bottom line.

Our next chapter will start a series of chapters on achieving the best value out of a BSM solution. First up will be a discussion on achieving management value out of the system. There, we'll further discuss the ROI associated with a BSM system. We'll dig deep into the management-level dashboards that provide decision-enabling data to business leaders. Most importantly, we'll come to understand some best practices in setting up visualizations for business leaders, and enterprise IT, as well as the unique requirements of service providers and outsourcers.

Download Additional eBooks from Realtime Nexus!

Realtime Nexus—The Digital Library provides world-class expert resources that IT professionals depend on to learn about the newest technologies. If you found this eBook to be informative, we encourage you to download more of our industry-leading technology eBooks and video guides at Realtime Nexus. Please visit <http://nexus.realtimepublishers.com>.