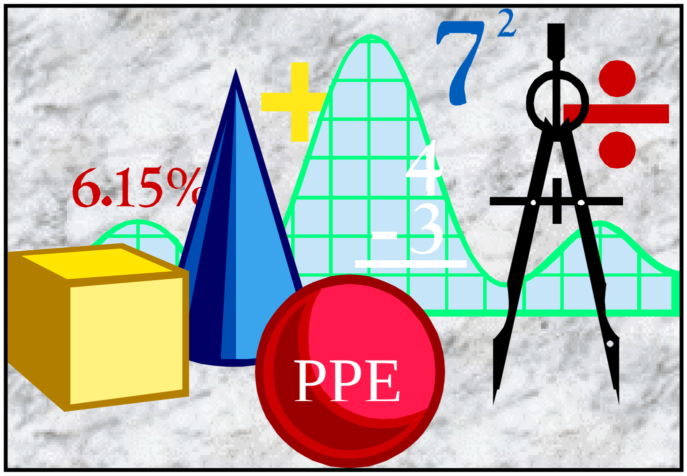


An Enterprise Level Approach to Proactive Performance Engineering

Authored by:
Michael K. Cook (Technical Specialist I, at State Farm Insurance)
michael.k.cook.aekf@statefarm.com

SES Conference On Performance Modeling - April 13-15, 1999

Date of last revision: Monday, March 15, 1999

Disclaimer

Table of Contents

Abstract

This is a paper on the experiences of selling and applying Proactive Performance Engineering (PPE) methods to proposed business system strategies at an enterprise level. PPE is a service that was introduced into our IT organization in January of 1997 in response to disappointing performance results observed from initial deployment of several distributed systems solutions. The goal of the PPE service was to provide a proactive approach to predictive performance modeling of distributed systems. Subjects included in this paper are the methods, tools, and techniques for establishing and selling a PPE service to the IT organization. Also included is a case study of applying PPE services at an enterprise level. Applying PPE at an enterprise level means focusing on the business system strategies targeted to key segments of the enterprise such as in an Insurance organization: Agency, Claims processing, Underwriting, and Call Center support. Experience has proven PPE services to be complementary to existing platform (i.e. operating system) based Capacity Planning processes. Since its inception in January 1997, PPE has emerged as a critical decision support service provided through all phases of the project development life cycle.

Getting Started

Introduction

PPE was chartered as a decision support service for Project Managers, Systems Designers/Architects, and Capacity Planners. The goal of the PPE service is to ensure quality performance is engineered into software applications and the system environments they are deployed. The PPE service was further structured and defined by incorporating best practices from several complimentary disciplines. PPE incorporates: Software Performance Engineering (SPE) [SMITH1990], [LOOSLEYDOUGLAS1998] discrete event systems modeling [SES], Monte Carlo based spreadsheet modeling [CRYSTAL BALL®], [RAGSDALE1998], analytic capacity planning [TANNER1995], and Design Of Experiments (DOE) for quality assurance and resource optimization [BREYFOGLE III1992].

Selling

PPE was not approved as a new service to the organization until successfully completing a "Proof Of Concept" pilot project and presenting the results bottom-up through the management structure. The ideas that played well in the management ranks included:

1. The potential return on investment by finding and fixing performance problems pro-actively.

2. Defining performance objectives for a project that can be measured and tested throughout the project development life cycle.

3. Gaining greater insight into the implementation of business solutions that spanned multiple system platforms, by asking questions necessary to building a predictive model.

4. Having the ability to model large segments of the enterprise (i.e. “Big Picture”) from the Mainframe systems down to the Desktop.

5. Predicting the “End-To-End” performance of tomorrow’s business solutions today.

PPE, once approved at the CIO level, was sold horizontally through the organization, first by targeting Project Managers then System Designers/Architects and Capacity Planners. Management support for funding the people resources to provide the service to the organization was key to the success of PPE. A core group of two internal resources initiated the service. Strategic partnerships with external associates accounted for as many as six additional resources to keep pace with the demand for the service.

Integration with Existing Services

Capacity Planning

Initially, there was some confusion over the difference between the existing platform based Capacity Planning process and PPE. This confusion was resolved by making the following distinctions.

PPE maintains a “Big Picture” view on proposed business solutions, which may span multiple systems and system platforms. PPE therefore, predicts the response time of business transactions from “End-To-End”. Capacity Planning focuses on existing workload trends, from a production environment and establishes capacity dates for a specific platform of systems based on internal response time measures.

PPE places an emphasis on performance early in the project life cycle (Pre-Production) to ensure systems are deployed to meet stated performance objectives, without disruption to existing business systems. Capacity Planning is an on going service that regularly checks the capacity dates of systems in production and schedules system upgrades.

PPE relies on the existing Capacity Planning process for baseline performance metrics from production systems to be included in enterprise PPE models.

Although it was originally intended that PPE would maintain a pre-production emphasis on predictive modeling, the number of requests to have a “Big Picture” business solution focus for production Capacity Planning has increased. These types of requests have resulted in the need to transition PPE models over to production Capacity Planners.

Performance Testing

In many organizations, performance testing if done at all, is usually the last and least complete testing done on a new system before it goes into production. Several reasons make performance testing an attractive option for cutting from the project schedule in favor of meeting implementation schedules.

Some of the more common arguments include:

- The business volume metrics are not known, like: number of concurrent clients, transaction mix, and transaction rates.
- Automated testing tools for performance testing distributed systems are difficult to use and maintain.
- Creating a physical testing environment of the enterprise is too costly or not a practical solution.
- Not enough qualified people resource to get the job done.

If your organization is currently doing some form of performance testing, then the need for proactive performance modeling may be questioned.

Performance modeling and performance testing are two activities that compliment each other – *they are not mutually exclusive*. Performance modeling provided through the PPE service can begin very early in the project life cycle. This early involvement will begin the *discovery process* that creates an awareness of performance issues. Issues like, performance goals and service level objectives for business transactions, transaction mix and rates, number of concurrent clients, and architecture of the system. All of these performance issues need to be addressed before one can proceed with a performance test plan and execution.

In addition, discovering significant performance defects during the performance test cycle is often too late to redesign or tune the business solution before it goes into production.

Delivering

As expected, selling the PPE service to the organization was a much easier task than delivering it. There were many lessons learned in our initial efforts. It was determined within the first 18 months of providing the PPE service that it worked better on some projects than it did on others. The following observations were made on successful efforts.

- Project leadership was mature and adhered to formal project methodologies.

- Volume metrics (e.g. client population, frequency and types of business transactions) were well understood and documented.

- The flow and frequency of a business transaction through the system was well understood and documented.

- The project had stated performance objectives.

- There was an awareness of performance issues and clearly stated performance questions that needed to be answered.

- The data volumes associated with a business transaction request were known or could be estimated.

- There was a good understanding of the proposed technologies and architecture that would be applied to the business requirements.

- There was a willingness to work with the PPE service provider to obtain model inputs and validate assumptions and constraints of the model.

- The PPE service was included early in the project life cycle to be a proactive effort.

- There was a good understanding of when important performance questions needed to be answered.

- A performance test plan and performance test execution was an integral part of the project.

A Wish-List of Enhancements

Having established the core services of PPE and assessing the effectiveness of the service, there are some enhancements that should be considered to provide added value. The enhancements are primarily targeted at conducting research and technology assessments to feed the results into projects that could take advantage of the up front research. This approach would also help to minimize the impact of PPE services on project time lines. A wish list of enhancements to consider are detailed below.

<u>Proactive Performance Research and Modeling</u>

Proactive performance research and modeling could be applied to leading edge technologies in both hardware and software components. The objective would be to use either an independent test laboratory or an internal test facility to develop performance profiles of the new technologies. If the preliminary research provided favorable results then model development and validation of the component(s) would be initiated.

Proactive Evaluation of Third Party Products

The use of third party products in many IT organizations is increasing at a rapid pace. The products are often acquired with only a cursory review of functionality and no forethought of performance, scalability, or impact on the infrastructure of the organization.

If the organization had a performance engineer involved prior to signing a contract there are several performance factors that could be addressed an incorporated into a contract, such as:

- If the product were performance engineered, then there would be documentation in place to back it up.
- Is there any evidence that the product has been tested for performance and scalability by an independent testing laboratory?
- If the product is a desktop application, how will the new application impact existing applications and system resources.
- When tailoring a third party product to your organization, specify a limit on system resource usage. This might prevent a costly upgrade to many systems in the organization, before deployment.
- How much of your existing network bandwidth will the new product be allowed to consume.
- Is the product consistent with the architecture of the organization?

Application functionality will only take you so far, if the product does not perform then clients will find away to work around using it.

Case Study

The case study presented in this paper has been generalized to protect the business initiatives and technology solutions of State Farm Insurance. However, the PPE methods being applied for an enterprise level approach are technically unchanged.

Initially, our PPE efforts were targeted at individual business initiatives that would be using a distributed system solution. This approach was good starting point for proving the concept of the PPE service. Once the PPE service was accepted, it became clear that there was a greater need to create a modeling "Blue Print" or "Big Picture" view of a key segment of the enterprise for which many business initiatives would be deployed over a period of four to five years.

Because of the overwhelming interest in this "Big Picture" view of modeling, the scope of the PPE service expanded to include an enterprise level approach.

Overview: Getting the "Big Picture"

The *discovery process* for obtaining the information required for the "Big Picture" began with identifying the business partner(s) that new this particular segment of the enterprise and the business strategies that would evolve over the next four to five years. Finding this business partner was the easiest part of the effort. Structuring the information that was needed as input for an enterprise level model was not as easy.

After some exploratory analysis of the business initiatives, it was decided that the information could be incorporated into a modeling spreadsheet that could also take advantage of the power of risk analysis techniques. This approach proved to be a very powerful and insightful first step to building a more complex *Strategizer®* model of the enterprise. The use of an "Add In" tool for spreadsheets from Decisioneering Inc., called Crystal Ball®, was used to provide the risk analysis capabilities. Crystal Ball® uses Monte Carlo simulation to forecast risk by using a deterministic spreadsheet model that has altered to incorporate stochastic assumption cells. The structure of an enterprise business planning and modeling spreadsheet are provided in the next section.

Enterprise Business Planning and Modeling Spreadsheet

The enterprise business planning and modeling spreadsheet used in this case study had six components, which include:

1. Enterprise metrics
2. Client metrics
3. Business initiatives or the application mix and usage
4. Key business transaction of an application mix
5. Critical resource estimates
6. "Back Office" server estimates

The spreadsheet is used to facilitate the discovery process for the "Big Picture". The objectives are to:

- Document the business volume metrics over multiple years
- Group the metrics into meaningful business initiatives
- Identify and focus on critical resources
- Quantify the "Back Office" transaction rates generated from the business initiatives
- Identify the business metrics that have variability, risk, uncertainty, or may be unknown
- Apply statistical distributions to quantify the variability, risk, uncertainty, or unknowns
- Identify key forecast metrics
- Use Monte Carlo simulation to explore the amount of risk associated with a forecast

The concept behind the spreadsheet assumes that the business organization has distributed systems environment consisting of multiple office tiers, which provide various levels of Information Technology support. The first tier represents the "Front Office" were client workstations and desktop applications reside. The second tier represents an integration layer to combine many first-tier offices into a centrally managed location. The second-tier may be referred to as the "Back Office" environment. The number of tiers will depend on the scope of the enterprise modeling effort. This case study is primarily a two-tier environment.

The spreadsheet was designed to allow for forecasting growth over multiple years. The volume metrics displayed in other sections of the spreadsheet is dependent on the selection entered in the "Year Index" in cell H4.

There are four types of cells used in the spreadsheet. The light yellow cells represent information provided from the business partner that are known values. The Green cells represent "assumption" cells that have variability, risk, uncertainty, or the "best guess" of an unknown. The blue cells are forecast cells that have been calculated for assessing risk. The white cells are simply calculated results.

The business volume metrics are calculated for both a workday as well as for a peak hour. The workday metrics are more meaningful to a business partner who will provided much of the input. Performance modeling and Capacity Planning efforts tend to focus on peak hour metrics; therefore, they were also included.

Enterprise Metrics

The first 15 rows of the spreadsheet shown in *Figure 1 - Enterprise Business Planning and Modeling Spreadsheet*, show the entries that comprise the enterprise metrics. The enterprise metrics serves to define the scope of the modeling effort. Insight into how the client population is distributed in a multi-tier organization is essential.

The business partner is expected to be able to complete the four metrics highlighted in yellow, on rows 5-6 and 8-9. The other enterprise metrics are either calculated directly or estimated by the performance engineer from input provided by the business partner. For example, the number of workstations active during the peak hour as opposed to the average over the workday would be estimated.

Figure 1 - Enterprise Business Planning and Modeling Spreadsheet

File Edit View Insert Format Tools Data Window Cell Run Help SES

	B	C	D	E	F	G	H	I	J	K
1	**ENTERPRISE BUSINESS PLANNING AND MODELING SPREADSHEET (V1.0)** DEVELOPED BY MICHAEL K. COOK (michael.k.cook.aekf@statefarm.com)									4:29:01 AM 3/4/99
2										
3		**Year End**					**Year Index**			
4	**Enterprise Metrics:**	**1999 (1)**	**2000 (2)**	**2001 (3)**	**2002 (4)**	<==========	4			
5	Employee Work Hours Per Day:	8	8	8	8					
6	Percent Active on Applications:	75%	75%	75%	75%					
7	Active Application Hours Per Work Day:	6	6	6	6					
8	Number of total first tier offices:	1,000	1,500	1,800	2,000					
9	Client workstations per office:	4	5	6	10					
10	Active Client Workstations Peak Hour:	4	5	6	10					
11	Active Client Workstations Work Day:	2	2.5	3	7					
12	Number of total client workstations:	4,000	7,500	10,800	20,000					
13	Number of total second tier offices:	8	8	8	8					
14	Average number of first tier offices per second tier:	125	188	225	250					
15	Average client workstations served per second tier:	500	938	1,350	2,500					
16							**Per First Tier Office (Work Day)**			
17	**Client Metrics:**						**TransPerHour**	**OneTrans EveryXSeconds**		
18	Client think time in seconds:	60	60	60	60		400.0	9.0		
19	Desired average response time seconds:	3	3	3	3					
20	Client transactions per Work Day:	343	343	343	343		**Per First Tier Office (Peak Hour)**			
21	Client transactions per Work Day per office:	686	857	1,029	2,400		**TransPerHour**	**OneTrans EveryXSeconds**		
22	Client transactions per Peak Hour:	57	57	57	57		571.4	6.3		
23	Client transactions per Peak Hour per office:	229	286	343	571					
24										
25										
26							**Per First Tier Office**			
27	**Business Initiatives Percent Of Time Spent On:**						**Volume Per Work Day**	**Volume Per Peak Hour**		**Minutes In Application**
28	Application 1:	75.000%	60.000%	50.000%	30.000%		720.00	171.429		108.000
29	Application 2:	8.000%	15.000%	30.000%	50.000%		1,200.00	285.714		180.000
30	Application 3:	5.000%	12.000%	12.000%	12.000%		288.00	68.571		43.200
31	Application 4:	5.000%	7.000%	5.000%	5.000%		120.00	28.571		18.000
32	Application 5:	4.000%	5.000%	2.000%	2.000%		48.00	11.429		7.200
33	Others:	3.000%	1.000%	1.000%	1.000%		24.00	5.714		3.600
34										
35	**Totals:**	**100.000%**	**100.000%**	**100.000%**	**100.000%**		**2,400.00**	**571.429**		**360.000**
36										
37										
38										
39										

BusinessMetrics / NetWorkMetrics / Application 2 / BackOfficeMetrics

Client Metrics

The Client Metrics serves to define business volume metrics for both an individual client and for a single first-tier Front Office.

The Client Think Time is defined as the amount of time a client consumes between screen or transaction entries while actively working in a business application.

The desired average response time is provided by the business partner to provide the performance engineer with the expected level of service.

Application Mix and Usage

The Application Mix and Usage section defines the business initiatives that will be deployed into the enterprise over a period of years. As you would expect, the business partner plays an essential role in providing this information. The design concept behind this section was to estimate potential transaction volumes from the various applications without the business partner having to labor over details of all the transaction types that could be generated.

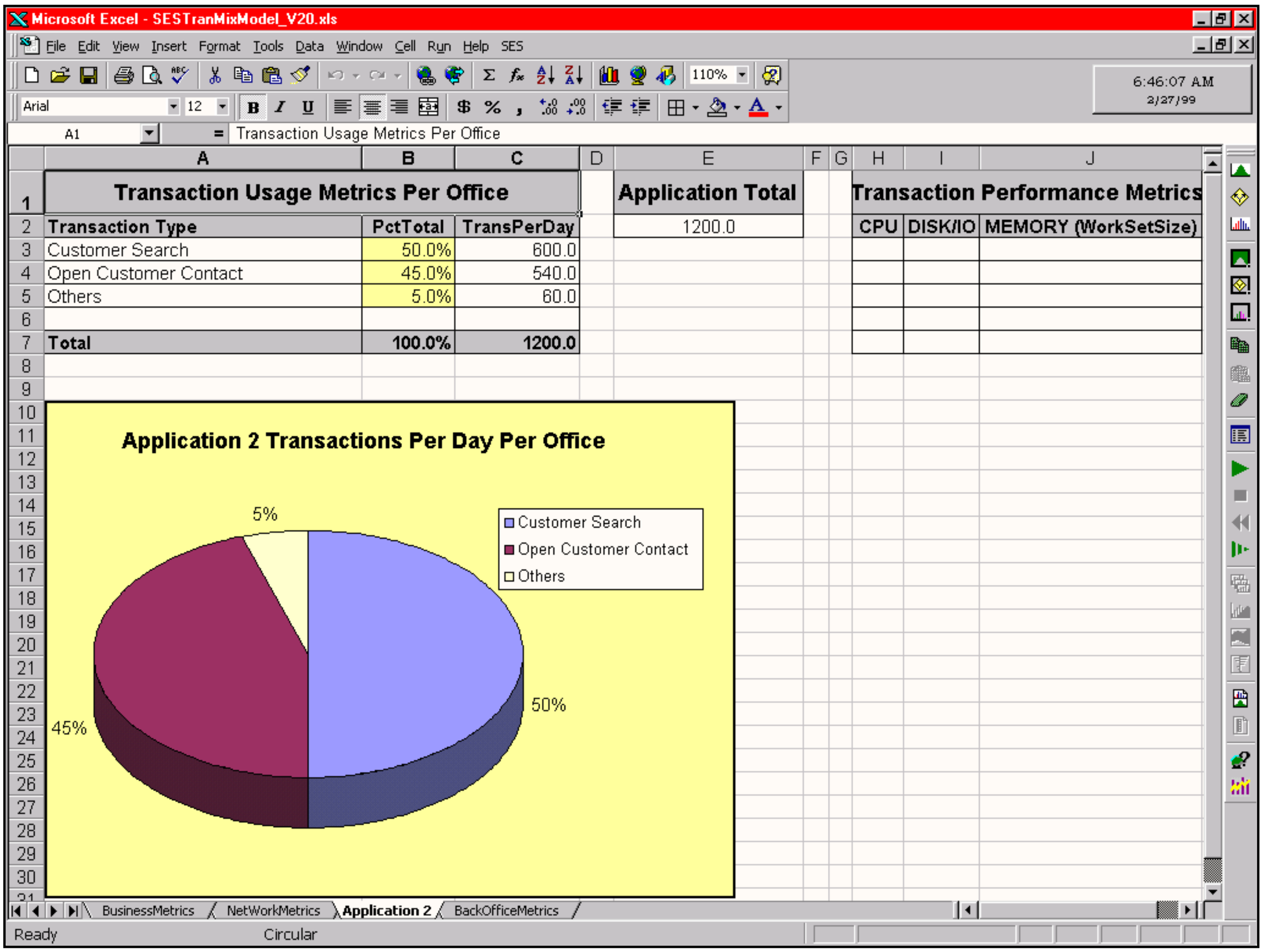
The business partner simply estimates the percentage of time in a workday that a client spends in an application. The percentage estimate provided by the business partner allows the corresponding transaction volume and usage minutes to be calculated. Information provided in the previous two sections: Enterprise Metrics, and Client Metrics, are used as input to the calculations in this section.

The business partner can use the “Minutes In Application” column as a sanity check for the “Percent of Time Spent On” estimate.

Key Business Transactions of an Application

A template for defining the key business transactions of an application is shown in *Figure 2 - Key Business Transactions*. Use the 80/20 rule to keep the number of application transactions to a manageable size. Once the key business transactions have been identified, the process of creating a performance profile of each can begin.

Figure 2 - Key Business Transactions



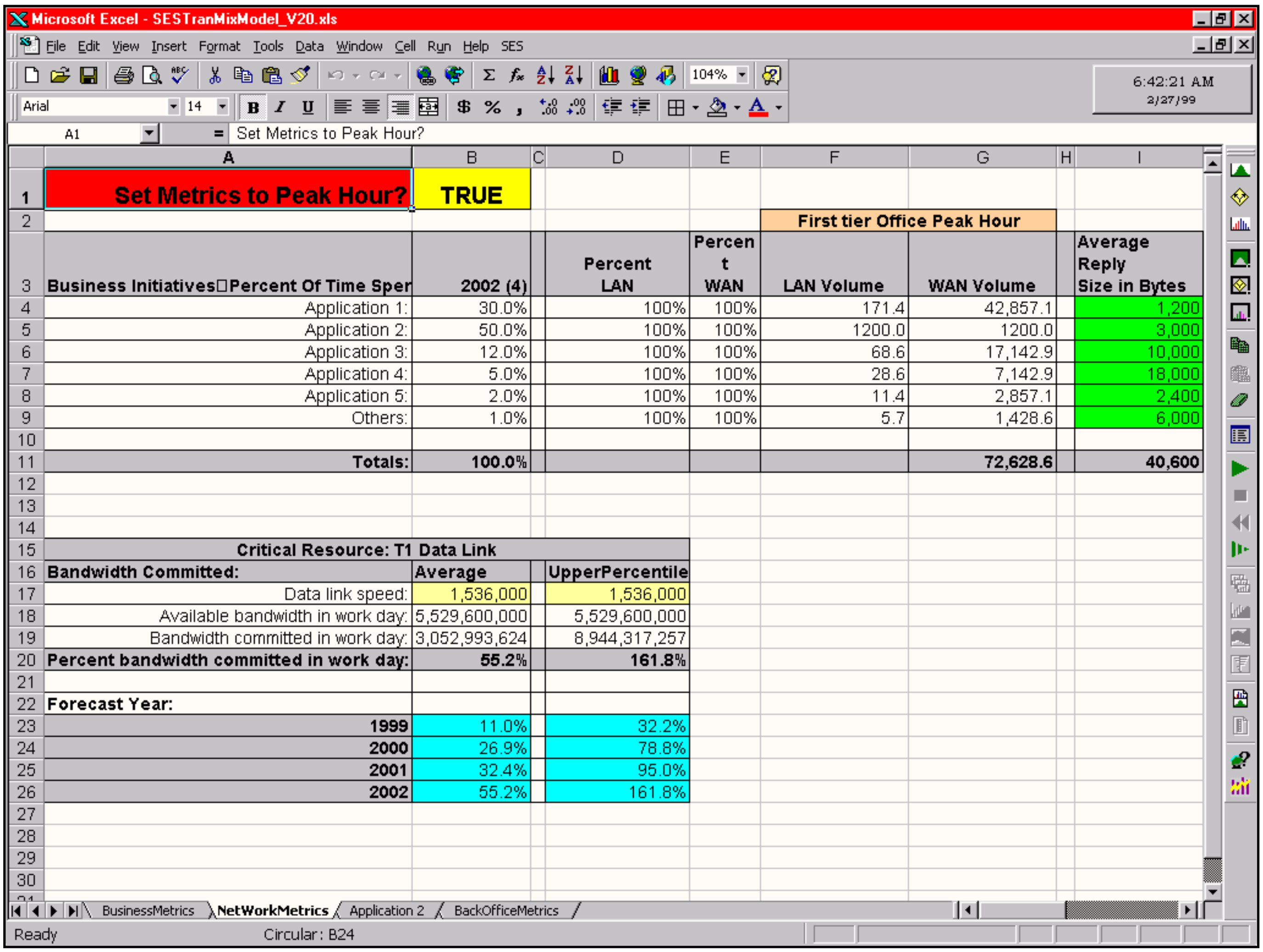
A performance profile includes: the frequency of occurrence for each transaction type, CPU path length, Disk I/O's, Memory Working Set Size, and any other pertinent information. The performance profile can be estimated initially from the "best guesses". The business partner should provide the in put on transaction frequency. An experienced performance engineer can provide the initial "best guesses" on the system metrics. Once a prototype of the application is developed, the "best guess" estimates can be refined. This refinement process iterates until a performance baseline test is completed on the final solution.

There are many benefits of having the key business transactions of an application identified. First, if test scripts will be written for a performance baseline of the application then the transactions that need to be scripted have been identified along with frequency counts. Secondly, it will provide the documentation needed to define which business transactions should be considered for "End-To-End" response time tracking in the production environment. Both of these checkpoints will provide valuable information for further validation of the enterprise modeling effort.

Critical Resources Estimates

In this case study, one of the critical resources identified is the single T1 Data Link that will be used to connect all 2,500 Front Office clients (i.e., Year 2002 estimate for 250 offices) to the Back Office. Because it is expected that the flow of data between the Front Office and Back Office applications will be asymmetrical, the message reply sizes from the Back Office to the Front Office are estimated for each application. The objective of this approach is to raise any "Red Flags" as early as possible.

Figure 3 - Critical Resource Estimates, Part 1



Microsoft Excel - SESTranMixModel_V20.xls

A1 = Set Metrics to Peak Hour?

	A	B	C	D	E	F	G	H	I
1	Set Metrics to Peak Hour?	TRUE							
2						First tier Office Peak Hour			
3	Business Initiatives□Percent Of Time Sper	2002 (4)		Percent LAN	Percent WAN	LAN Volume	WAN Volume		Average Reply Size in Bytes
4	Application 1:	30.0%		100%	100%	171.4	42,857.1		1,200
5	Application 2:	50.0%		100%	100%	1200.0	1200.0		3,000
6	Application 3:	12.0%		100%	100%	68.6	17,142.9		10,000
7	Application 4:	5.0%		100%	100%	28.6	7,142.9		18,000
8	Application 5:	2.0%		100%	100%	11.4	2,857.1		2,400
9	Others:	1.0%		100%	100%	5.7	1,428.6		6,000
10									
11	Totals:	100.0%					72,628.6		40,600
12									
13									
14									
15	Critical Resource: T1 Data Link								
16	Bandwidth Committed:	Average		UpperPercentile					
17	Data link speed:	1,536,000		1,536,000					
18	Available bandwidth in work day:	5,529,600,000		5,529,600,000					
19	Bandwidth committed in work day:	3,052,993,624		8,944,317,257					
20	Percent bandwidth committed in work day:	55.2%		161.8%					
21									
22	Forecast Year:								
23	1999	11.0%		32.2%					
24	2000	26.9%		78.8%					
25	2001	32.4%		95.0%					
26	2002	55.2%		161.8%					

BusinessMetrics | NetWorkMetrics | Application 2 | BackOfficeMetrics

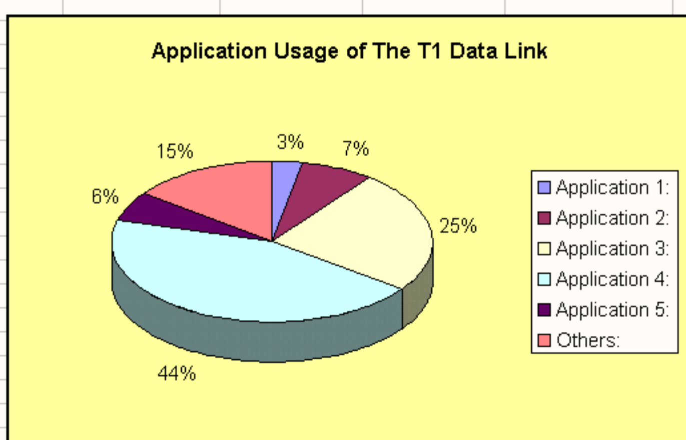
Ready Circular: B24

Figure 4 - Critical Resource Estimates, Part 2, provides a visual assessment of which applications are using most of the critical resource – the T1 Data Link in this case.

Figure 4 - Critical Resource Estimates, Part 2

Microsoft Excel - SESTranMixModel_V20.xls

					First tier Office Peak Hour		
Row	Average Reply Size in Bytes	Average Reply Size in Bytes	(Average Reply Size in Bytes + OverHead) *3	Percentage of Total Bytes	Total Bits	(Total Bits)*3	BitsPerSecond
4	1,200	1,207	3,622	3.0%	423,830,674	1,241,691,429	114,971.4
5	3,000	3,018	9,054	7.4%	29,668,147	86,918,400	1,373.5
6	10,000	10,060	30,180	24.6%	1,412,768,914	4,138,971,429	383,238.1
7	18,000	18,108	54,324	44.3%	1,059,576,686	3,104,228,571	287,428.6
8	2,400	2,414	7,243	5.9%	56,510,757	165,558,857	15,329.5
9	6,000	6,036	18,108	14.8%	70,638,446	206,948,571	19,161.9
10							
11	**40,600**	**40,844**	**122,531**		**3,052,993,624**	**8,944,317,257**	**821,503.0**



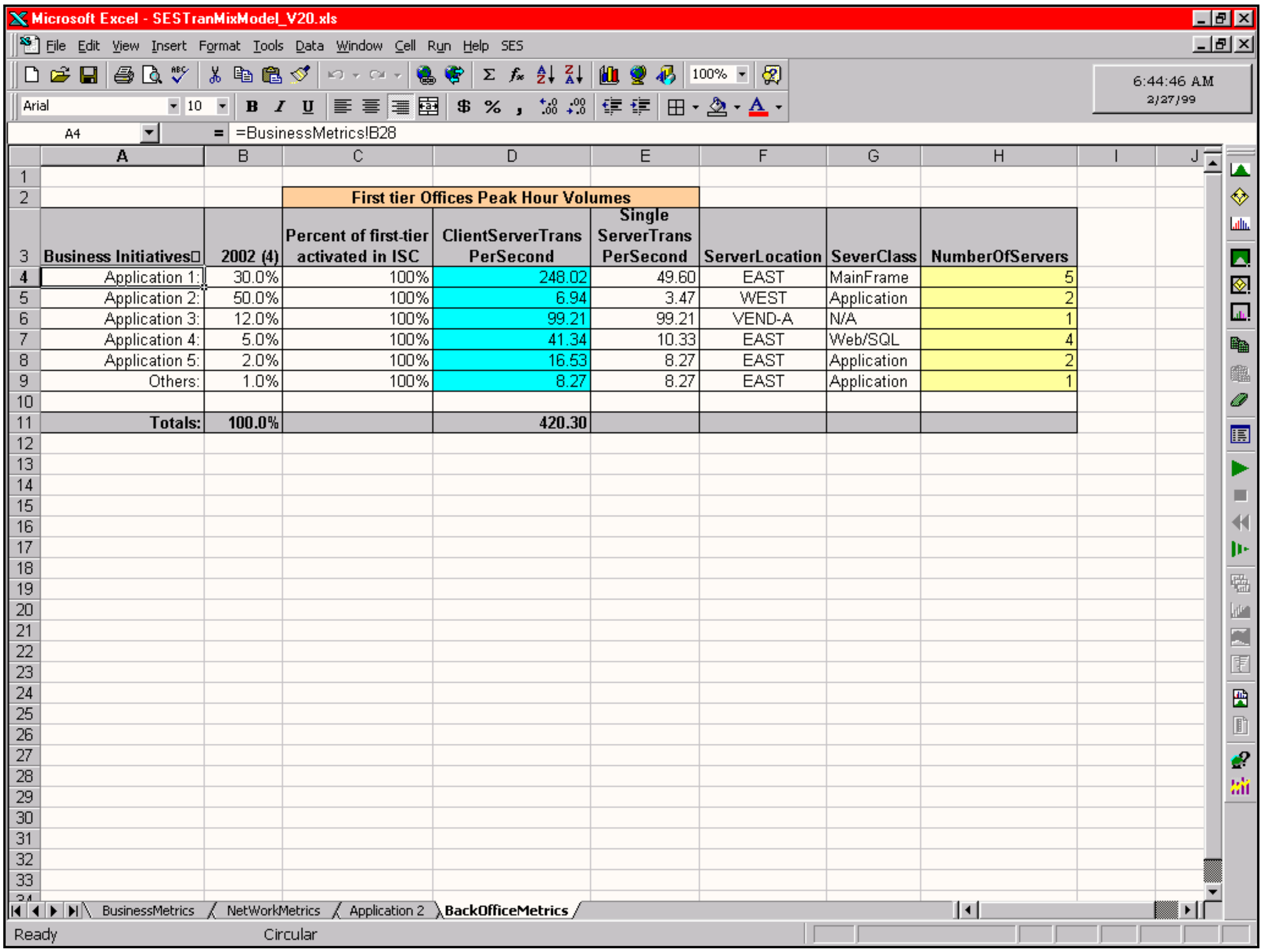
BusinessMetrics | NetWorkMetrics | Application 2 | BackOfficeMetrics

Ready   Circular: B24

“Back Office” Server Metrics

The Back Office Server Metrics section was developed to provide the performance engineer with an early estimate of the rate at which client transactions from the Front Office will arrive to servers applications. This section could be easily enhanced to also provide an estimate on the number of servers that may be needed to support the client population demand.

Figure 5 - Back Office Server Estimates



Business Initiatives	2002 (4)	Percent of first-tier activated in ISC	ClientServerTrans PerSecond	Single ServerTrans PerSecond	ServerLocation	SeverClass	NumberOfServers
		First tier Offices Peak Hour Volumes					
Application 1:	30.0%	100%	248.02	49.60	EAST	MainFrame	5
Application 2:	50.0%	100%	6.94	3.47	WEST	Application	2
Application 3:	12.0%	100%	99.21	99.21	VEND-A	N/A	1
Application 4:	5.0%	100%	41.34	10.33	EAST	Web/SQL	4
Application 5:	2.0%	100%	16.53	8.27	EAST	Application	2
Others:	1.0%	100%	8.27	8.27	EAST	Application	1
Totals:	**100.0%**		**420.30**				

Strategizer®

<u>Transition from Spreadsheet to *Strategizer®*</u>

With the Enterprise Business Planning and Modeling spreadsheet complete, making the transition to *Strategizer®* is a much easier task. I found that representing the physical environment first and then incorporating the framework for the application and business transactions mix was good approach.

The *Strategizer®* model of the physical environment is shown in *Figure 6 - Case Study Strategizer® Model*. The model has the “Front Office” as the first-tier where the client workstations and desktop applications reside. The “Back Office” second-tier has a small web-farm, one mainframe system, and a small application-farm.

The first objective is to build a framework from which the appropriate mix of applications and their corresponding business transactions can be generated. Once the framework is in place, the model can be tested to determine if the application mix and business transaction rates validate to the spreadsheet. This first objective is a big step toward building a model that reflects the enterprise business environment.

<u>Building the Application and Business Transaction Mix Framework</u>

Controlling the application and business transaction mix is accomplished by defining a set of constants, which specify the frequency or probability of transaction being generated from a particular application. The constants used in the case study are shown in *Figure 7 - Application and Business Transaction Mix*.

I first defined the constants that control the probability of an application being used by the clients. These constants are labeled as cMix_Application#. Each application has a cumulative probability value. I say “cumulative” because you will notice that *Figure 1 - Enterprise Business Planning and Modeling Spreadsheet*, shows for year ending 2002 that Application2 is used 50% of the time. However, in the *Strategizer®* model the value entered in cMix_Application2 is 80%. The reason for the discrepancy is that cMix_Application1 was first entered as 30%. This approach will better understood when the scripting for the *Strategizer®* behavior is reviewed in the section: <u>Creating Behavior Scripts for the Application and Business Transaction Mix Framework</u>

The same approach is used for defining the constants to control the probability of a business transaction being entered by a client from a particular application. The constants are labeled as cMix_Appl#_*BusinessTransactionName*. For example cMix_Appl2_CustSearch has the probability value for the Customer Search transaction from Application number 2.

Figure 6 - Case Study Strategizer® Model

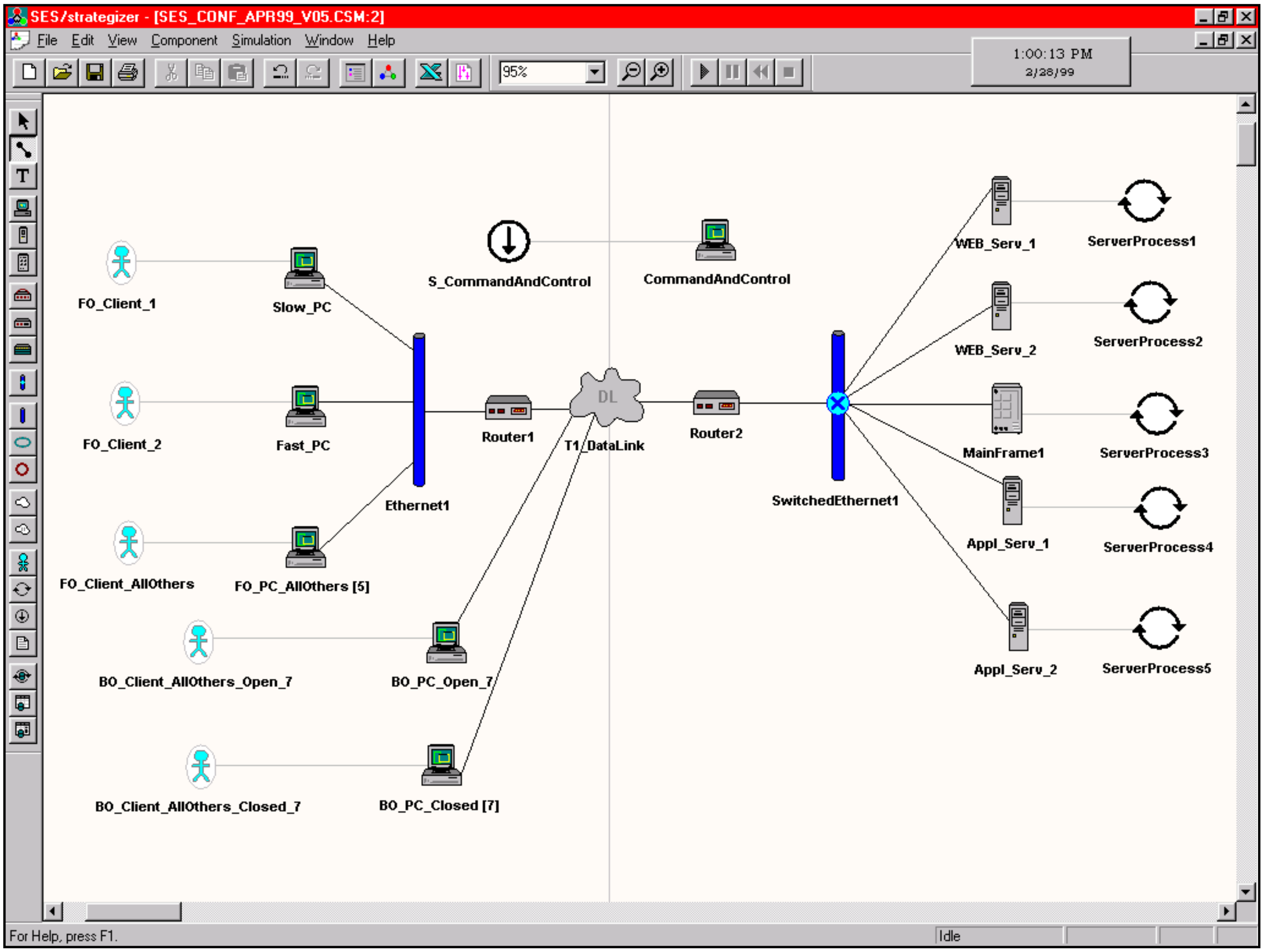
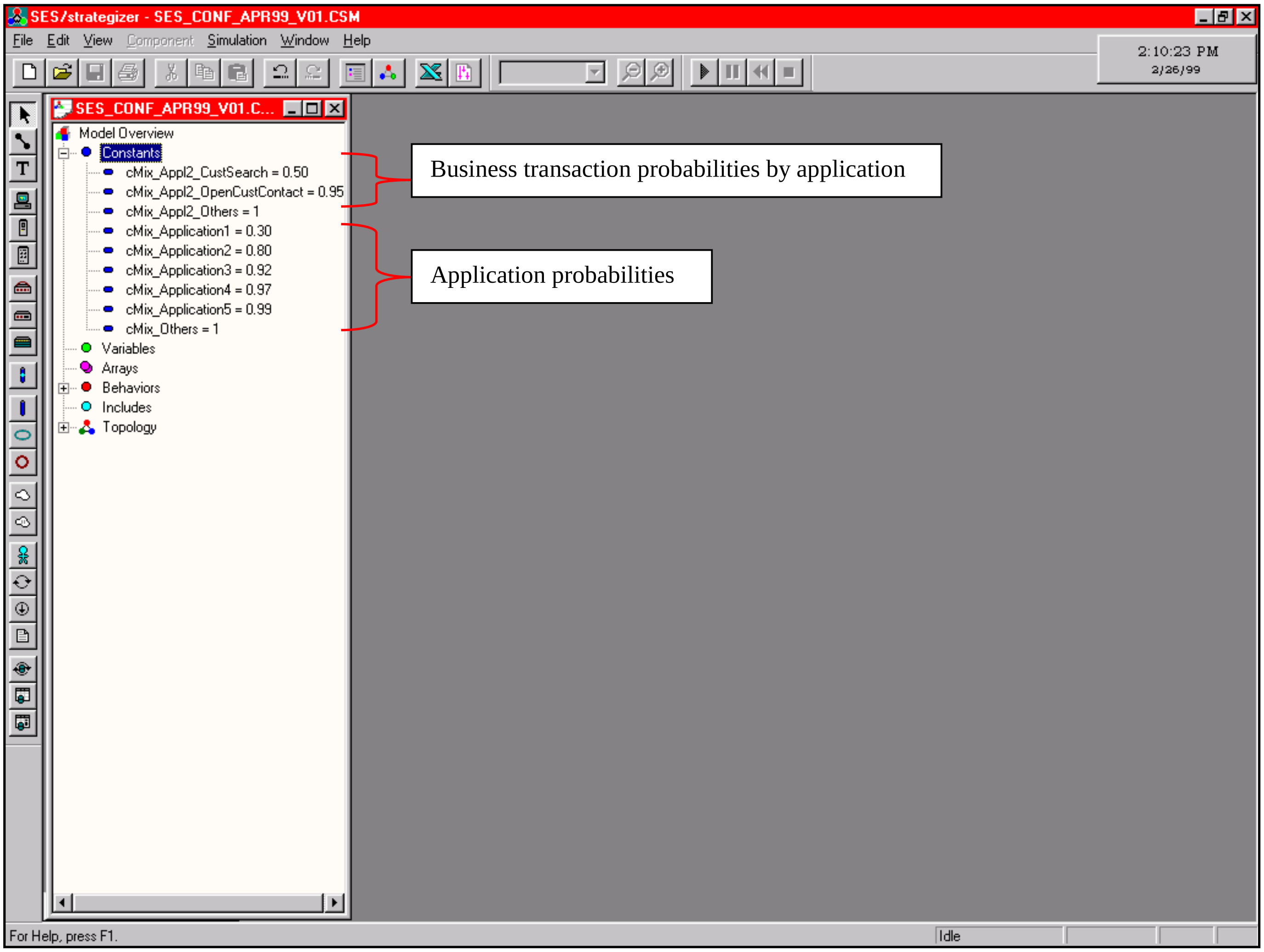


Figure 7 - Application and Business Transaction Mix



Creating Behavior Scripts for the Application and Business Transaction Mix Framework

In the case study model, the business transactions for each application were implemented as Server Process behaviors. To simplify building the framework behaviors, you should first setup the Server Process behaviors. The setup for this is shown in *Figure 8 - Server Process Behavior Scripts*.

After the Server Process behaviors have been defined, the implementation of the framework is a simple three-step process, Starting with *Figure 9 - Front Office Client 1 Dialogue* and ending with *Figure 11 - Client Transaction Mix Script for Application 2*. The process is shown as a series of screen shots which demonstrate how *Strategizer®* was used to build the framework behavior scripts for the application and business transaction mix.

Figure 8 - Server Process Behavior Scripts

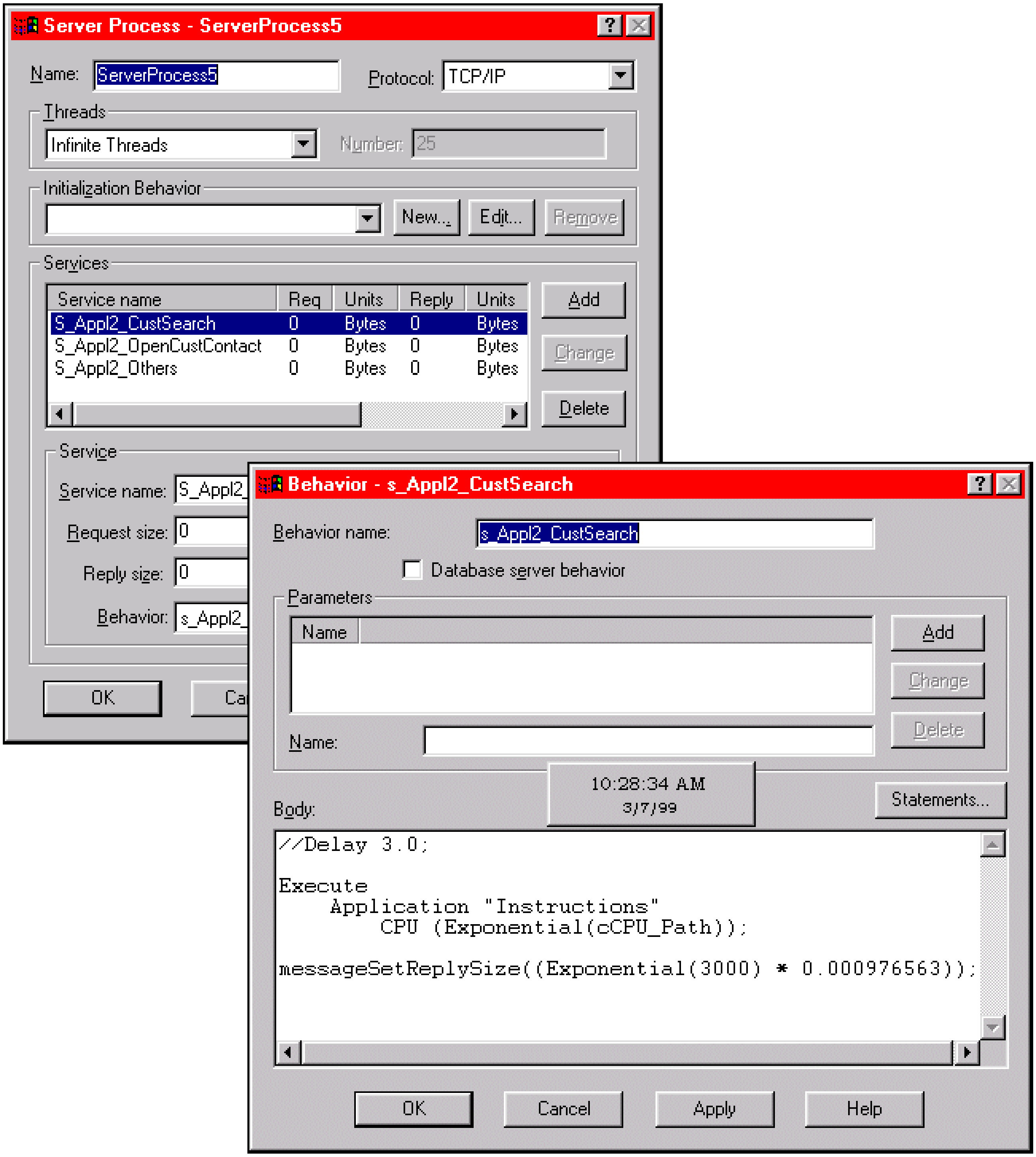
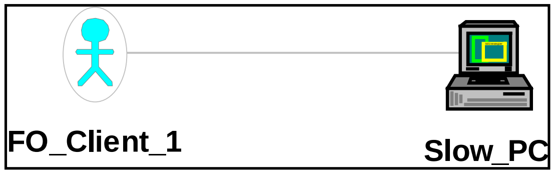


Figure 9 - Front Office Client 1 Dialogue



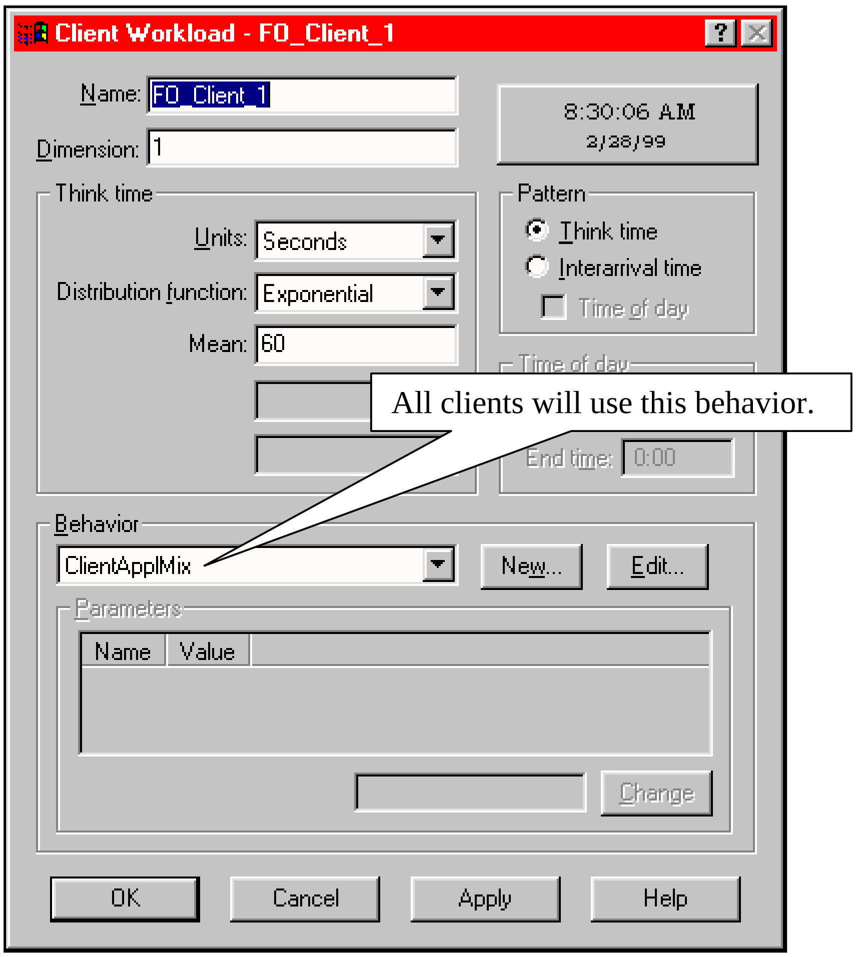


Figure 10 – Client Application Mix Script

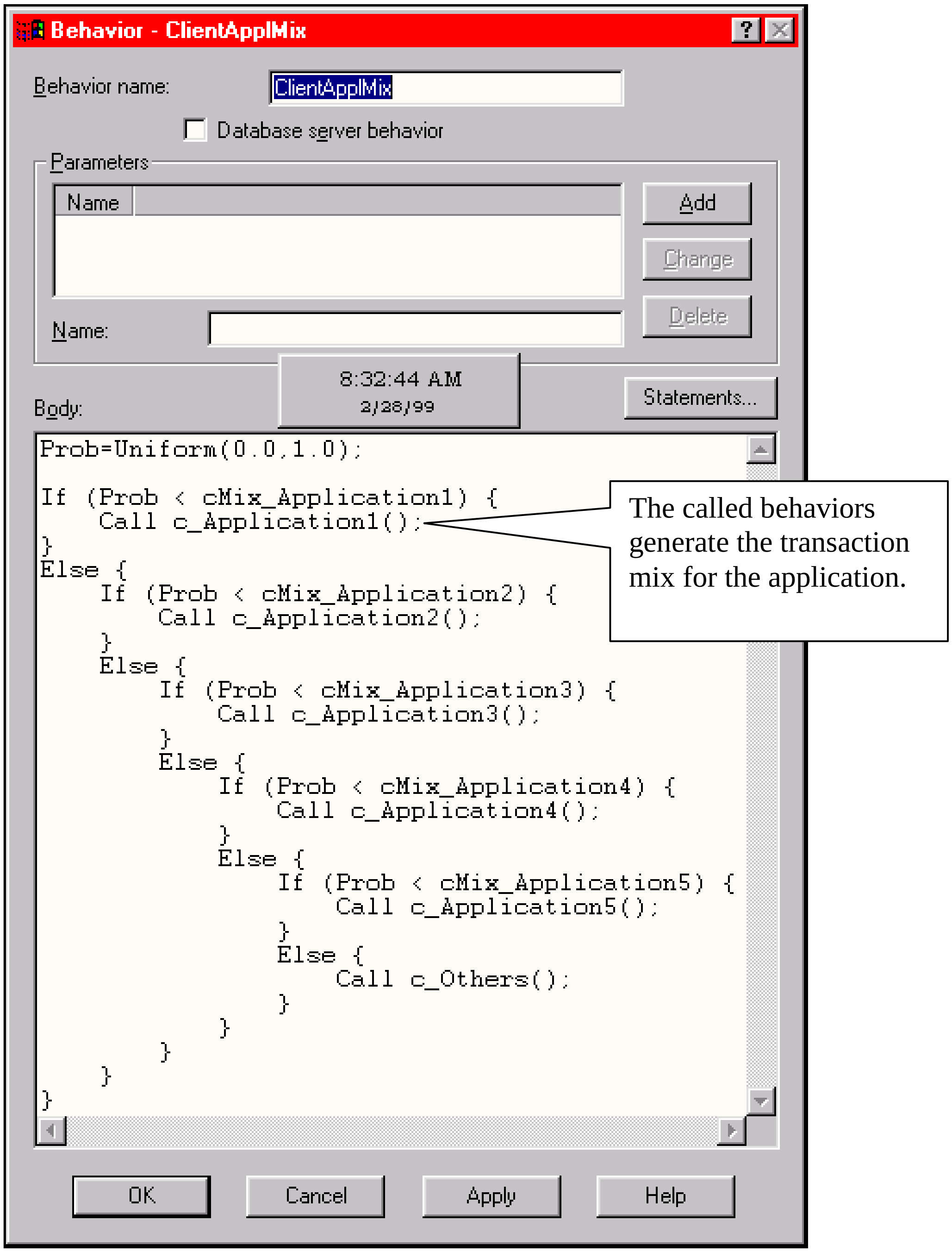
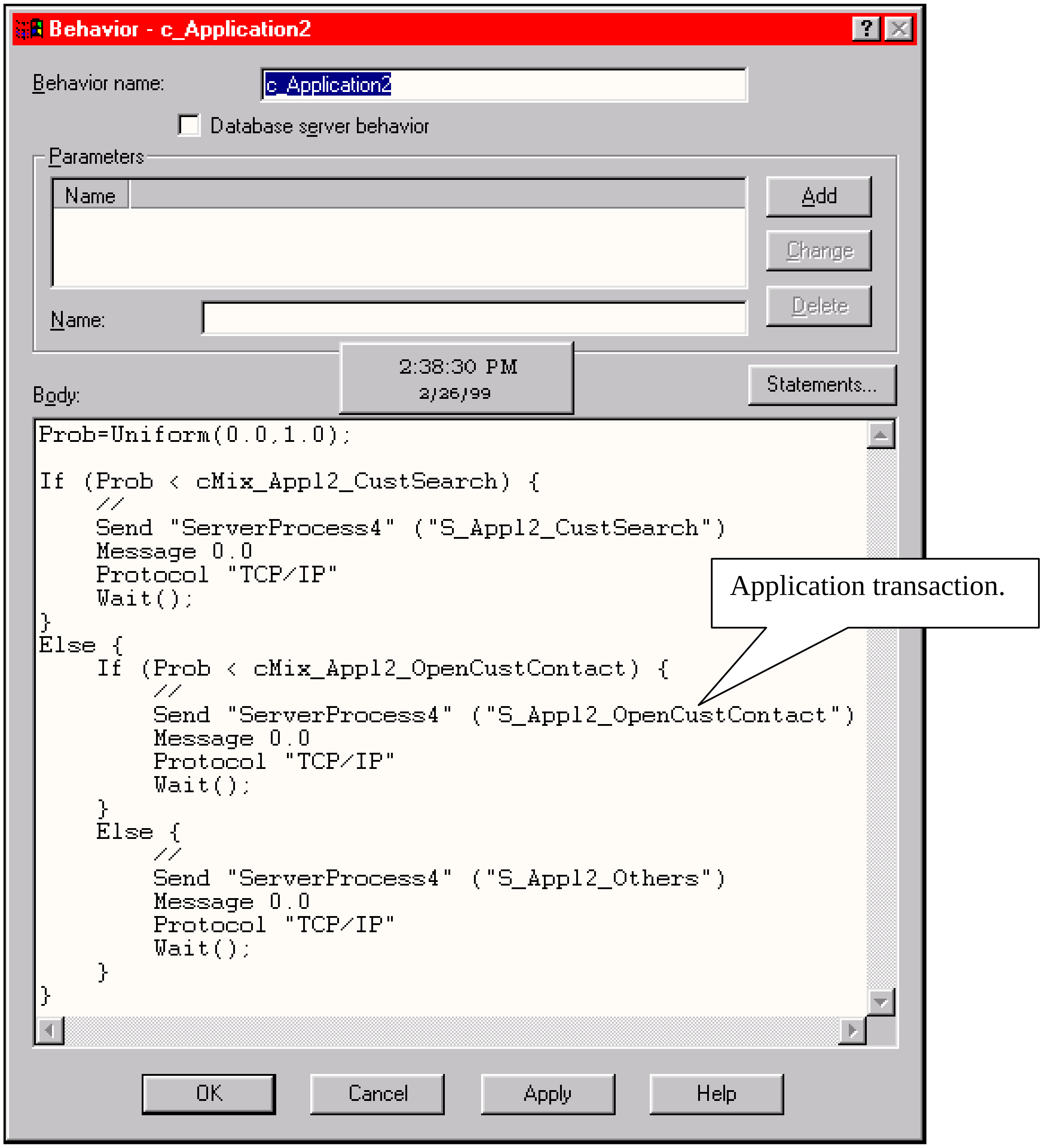


Figure 11 - Client Transaction Mix Script for Application 2



Dealing with the “AllOthers” Client Workloads

There are several methods of representing the workloads of all other clients. The quickest approach is to apply background load to all resources used by the other clients. A second approach is to dimension the client workstation to the client population size. Dimensioning the client workstation works fine as long as the size is in the hundreds or less. A third approach I’ve used to represent client populations in the thousands is to divide the think time used for a single client by the “AllOthers” size. For example, if the think time for a single client is 60 seconds and the size of “AllOthers” clients is 1,000, then the new time value is 0.06 seconds. The new time value however must be used as an *Interarrival* time in the client dialogue box rather than a *Think* time. One must realize that substituting think time with an interarrival time will change the client model from a “Closed” or finite population to an “Open” or infinite population.

When substituting an Open model for a Closed one, the Open model will tend to generate a slightly higher transaction rate. The higher transaction rate is observed because, in an Open model the initiation of transactions by the client behavior is independent of the time required to complete any previous transactions. Calibration of the Open model approach may be needed, if the transaction rates are beyond an acceptable level. The calibration can be accomplished by adding a baseline average transaction response time to the think time, before dividing think time by the size of “AllOthers”. The Open and Closed modeling techniques will be contrasted in the next section on Model Validation.

Simulating client populations in the thousands is only done to establish a baseline of resource usage that can be used for assigning background load values for subsequent simulations. Using this technique will significantly speedup the simulation execution elapsed time.

Model Validation

With the framework in place for controlling the application and business transaction mix, the case study model can now be validated for transaction rates or volumes, using the Enterprise Business Planning and Modeling spreadsheet. In addition, the differences between the Open and Closed modeling techniques can be explored.

For validation of the case study, the model was configured with three first-tier Front Offices. There are seven clients in each Front Office. The dialogue boxes for the seven clients configured as an Open model are shown in *Figure 12 - Open Client Configuration for AllOthers*. The dialogue boxes for the seven clients configured as Closed model are shown in *Figure 13 - Closed Client Configuration for AllOthers*. The results from the validation test run are shown in *Figure 14 - Validation Results*.

Figure 12 - Open Client Configuration for AllOthers

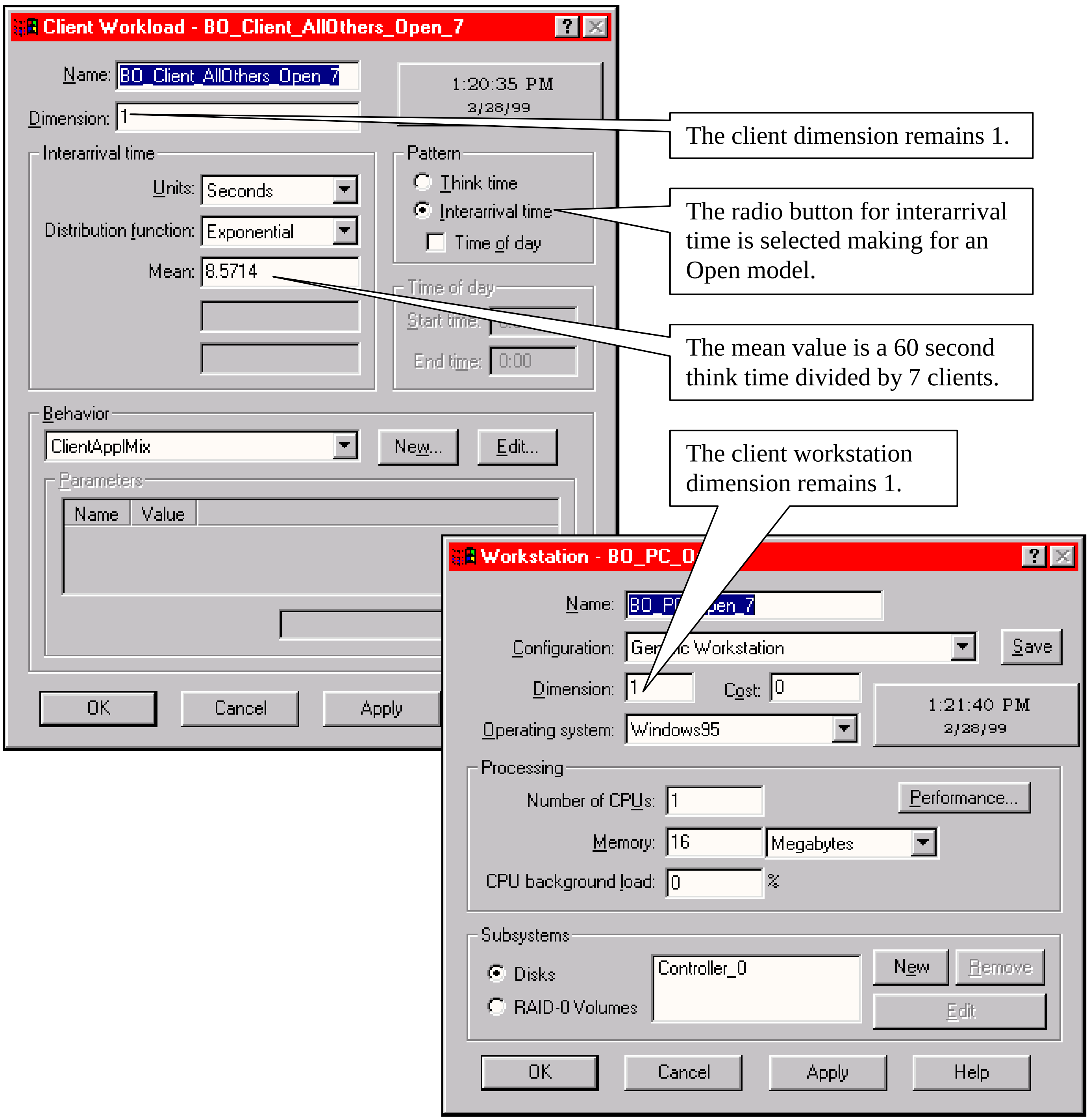


Figure 13 - Closed Client Configuration for AllOthers

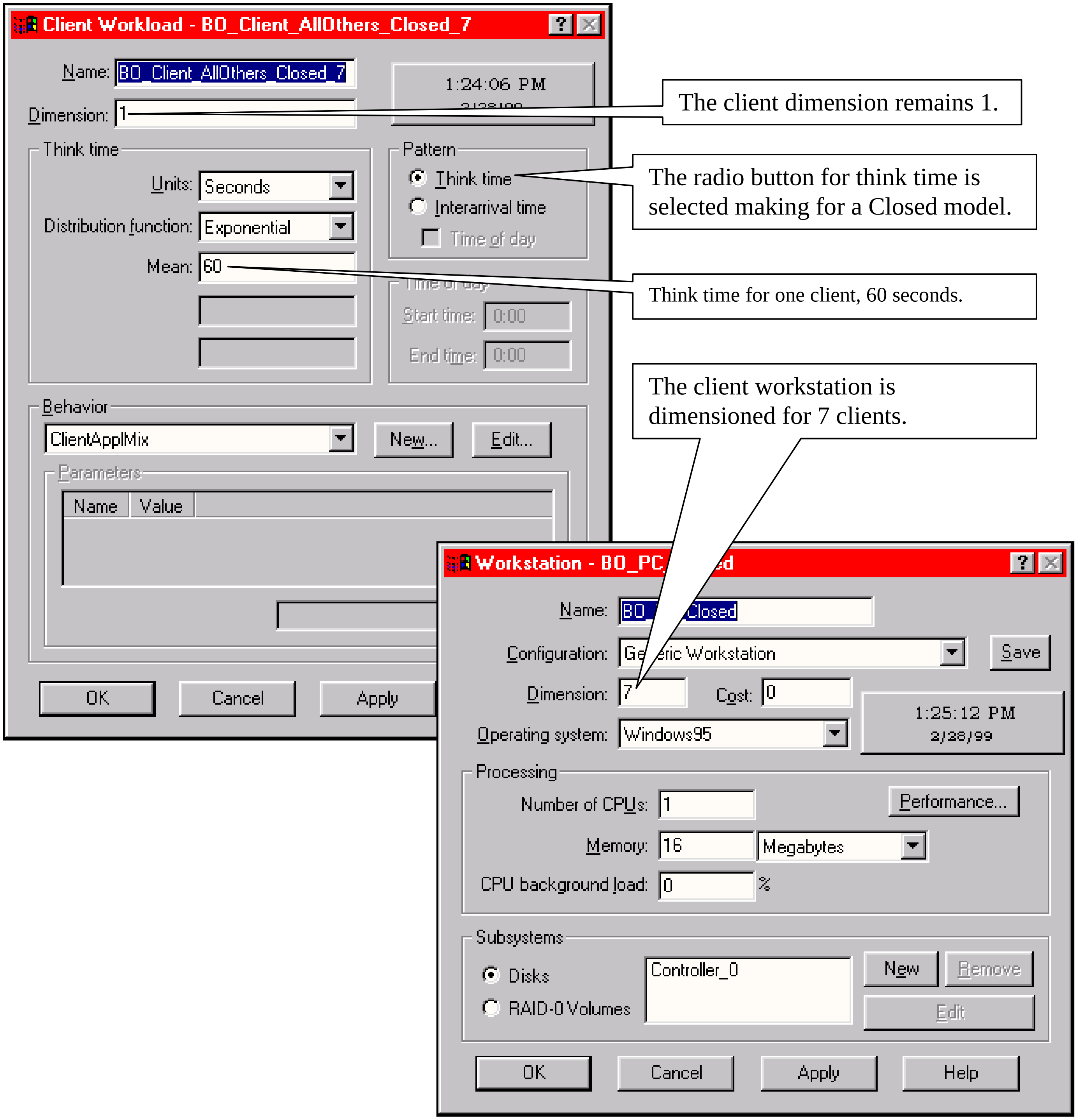


Figure 14 - Validation Results

Volume Per Work Day
720.00
1,200.00
288.00
120.00
48.00
24.00
2,400.00

Transaction Usage Metrics Per Office

Transaction Type	PctTotal	TransPerDay
Customer Search	50.0%	600.0
Open Customer Contact	45.0%	540.0
Others	5.0%	60.0
Total	**100.0%**	**1200.0**

File Edit View Insert Format Tools Data Window Cell Run Help SES

	A	B	F	G	H	I	J	K
1	Model results "SES_CONF_APR99_V05" run on Sun Feb 28 13:28:49 1999 for 21600 seconds (after 0 second warm-up).							
2								
3	Run Description							1:45:53 PM 2/28/99
4								
5								
6	Parameters							
7								
8								
9								
10	Client Workload Response Time Statistics (seconds)							
11		*Client Workload*	*Mean*	*Deviation*	*Minimum*	*Maximum*	*Sample Count*	
12		BO_Client_AllOthers_Closed_7	2.94	3.23	0.0310	53.8	2462	
13		BO_Client_AllOthers_Open_7	2.94	3.28	0.0310	64.4	2494	
14		FO_Client_1	2.90	3.23	0.0312	40.9	321	
15		FO_Client_2	2.81	3.40	0.0311	42.4	343	
16		FO_Client_AllOthers	2.82	3.03	0.0311	37.8	1659	
17								
18	Client Workload Interarrival Time Statistics (seconds between initiations)							
19		*Client Workload*	*Mean*	*Deviation*	*Minimum*	*Maximum*	*Sample Count*	*Arrival rate*
20		BO_Client_AllOthers_Closed_7	61.3	58.4	0.229	542.	2455	0.0162
21		BO_Client_AllOthers_Open_7	8.66	8.64	0.00168	77.6	2492	0.115
22		FO_Client_1	67.1	63.0	0.991	401.	319	0.0148
23		FO_Client_2	63.0	60.1	0.829	335.	341	0.0158
24		FO_Client_AllOthers	65.0	62.8	0.975	639.	1654	0.0153
25								
26	Behavior Response Time Statistics (seconds)							
27		*Behavior*	*Mean*	*Deviation*	*Minimum*	*Maximum*	*Sample Count*	Normalized
28		ClientApplMix	2.90	3.21	0.0310	64.4	7279	**2,426**
29		c_Application1	3.27	1.41	2.43	11.0	2220	**740**
30		c_Application2	2.78	4.38	0.0310	64.4	3630	**1,210**
31		c_Application3	2.74	0.658	2.43	6.66	841	**280**
32		c_Application4	2.55	0.139	2.43	3.61	363	**121**
33		c_Application5	2.46	0.111	2.43	3.20	148	**49**
34		c_Others	2.48	0.0796	2.43	3.06	77	**26**
35		s_Appl2_CustSearch	2.61	2.69	0.000723	23.2	1765	**588**
36		s_Appl2_OpenCustContact	2.88	5.82	0.000000	64.4	1676	**559**
37		s_Appl2_Others	2.61	0.508	1.72	5.05	189	**63**
38		s_Application1	3.23	1.41	2.40	11.0	2220	
39		s_Application3	2.66	0.655	2.40	6.56	841	
40		s_Application4	2.42	0.0934	2.40	3.26	363	
41		s_Application5	2.42	0.109	2.40	3.13	148	

SES_CONF_APR99_V05 / Hardware Utilization Details

The case study model was run for a total of 6 hours (simulation time) to obtain the transaction volumes from a workday as was originally estimated back in *Figure 1 - Enterprise Business Planning and Modeling Spreadsheet*.

The two item labels 1 & 2 in *Figure 14 - Validation Results*, compare the application mix transaction volumes from the enterprise business planning and modeling spreadsheet with those results obtained from the *Strategizer®* case study model. The results are certainly within reason considering the former is a deterministic model and the latter a stochastic model. Likewise, the business transaction volumes for Application 2, as indicated by item label 3 above, are also within reason. Note that the transaction volumes for Application 2 have been normalized to one Front Office (i.e., divided by 3) to compare back to the volumes originally shown in *Figure 2 - Key Business Transactions*.

Item label 4 compares the Closed model client transaction volumes to those of the Open model. Notice how the Open model produces slightly more transactions than the Closed. If the difference is too large, you may want to increase the *Interarrival* time on the Open model to reduce the number of transactions. Because a baseline average response time (Mean value in item label 4) is now available from a test run of the case study model, the calculation for calibrating the Open model client interarrival time is now:

(60 <u>seconds of think time</u> + 2.94 <u>seconds of response time</u>) / 7 <u>clients</u>, or 8.99 seconds.

Take note that if changes to the model significantly change the average response time of the transactions, the interarrival time for the Open model client will need to be re-calibrated.

Presenting Response Time Results

One of the more effective graphics used by the PPE group to report simulation test results back to the business partner has been the Box & Whisker chart. The Box & Whisker (B&W) charts are used to visually show and compare response times of the key business transactions. Prior to using a Box & Whisker chart for presenting results, you will need to level set your audience on how to interpret the chart. The B&W charts have been accepted in our organization because they provide greater insight into the quality of business transaction response times as well as an indication of what the client perceived response time would be. The graphics presented in this section were produced from *Statistica®*, a product of StatSoft, Inc.

Figure 15 - Compare Bar Graph to Box & Whisker Chart of Response Times, provides a direct comparison of reporting response times using a bar graph Vs. the B&W chart. The bar graph simply shows the mean value response time of the business transaction from Application 2 of the case study model. From the bar graph, one might be mislead to believe that there is not a significant difference between the response times of the three transactions. The B&W chart however, reveals quite a different story about the response times of the three transactions.

First, the mean value reported in the bar graph would indicate that the measure of central tendency is approximately 2.5 – 3.0 seconds. The B&W chart shows, by reporting the median value that the true measure of central tendency is very different depending on the business transaction. The median will be between 0.5 and 2.5 seconds.

Secondly, because the bar graph shows only the mean value response time the amount of variability is unknown. The B&W chart shows variability in terms of percentiles. It is obvious to see that the response times of the "Other" transaction group provide a better quality level of service with more consistent responses. A client is likely to say that responses of the "Open Contact" group are unacceptable. It is the clients perception of response time that becomes the overriding issue. Therefore, greater attention needs to be placed on what the clients perceptions of response time will be, not a statistical average.

Figure 15 - Compare Bar Graph to Box & Whisker Chart of Response Times

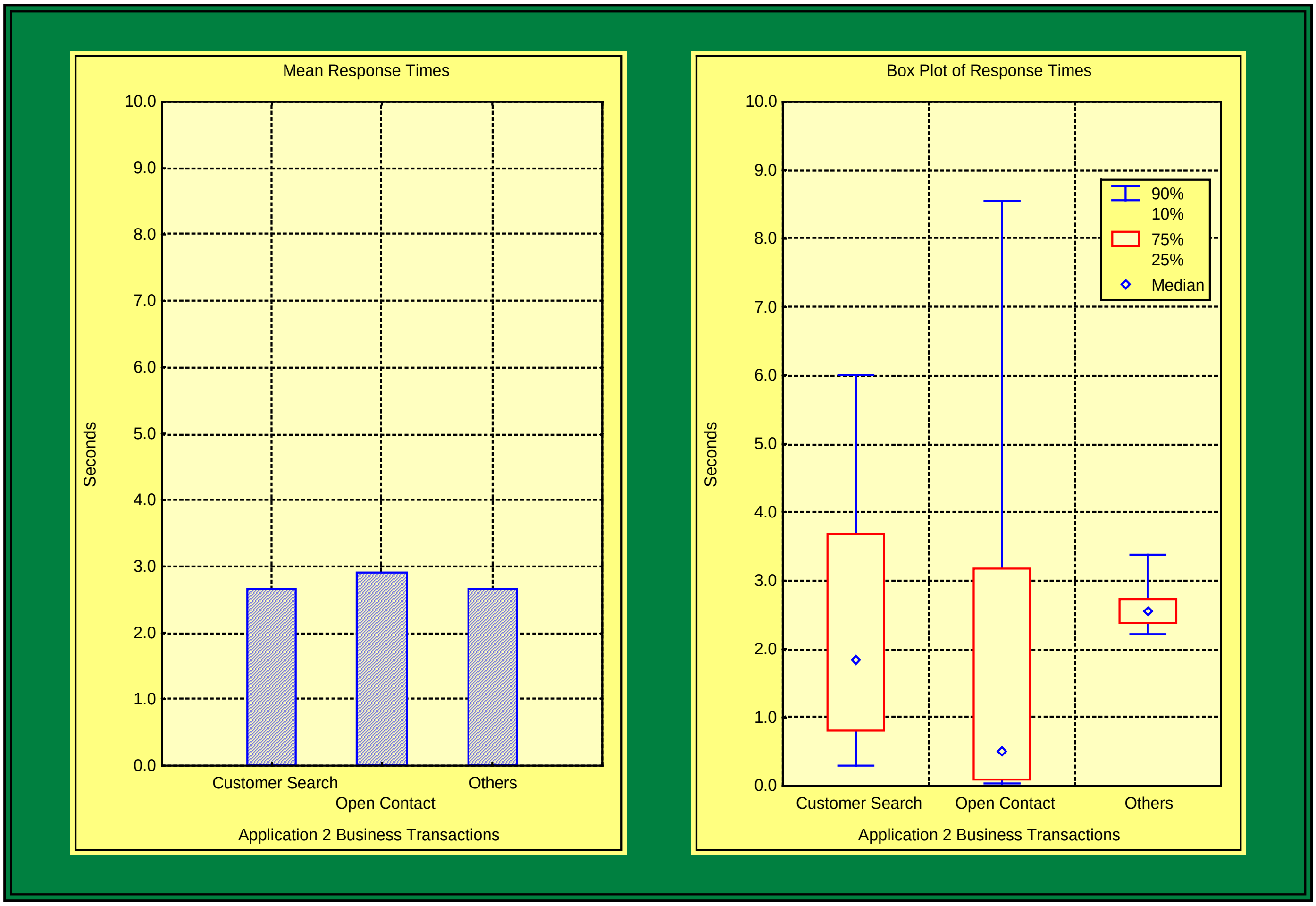


Figure 16 - Response Time Perception

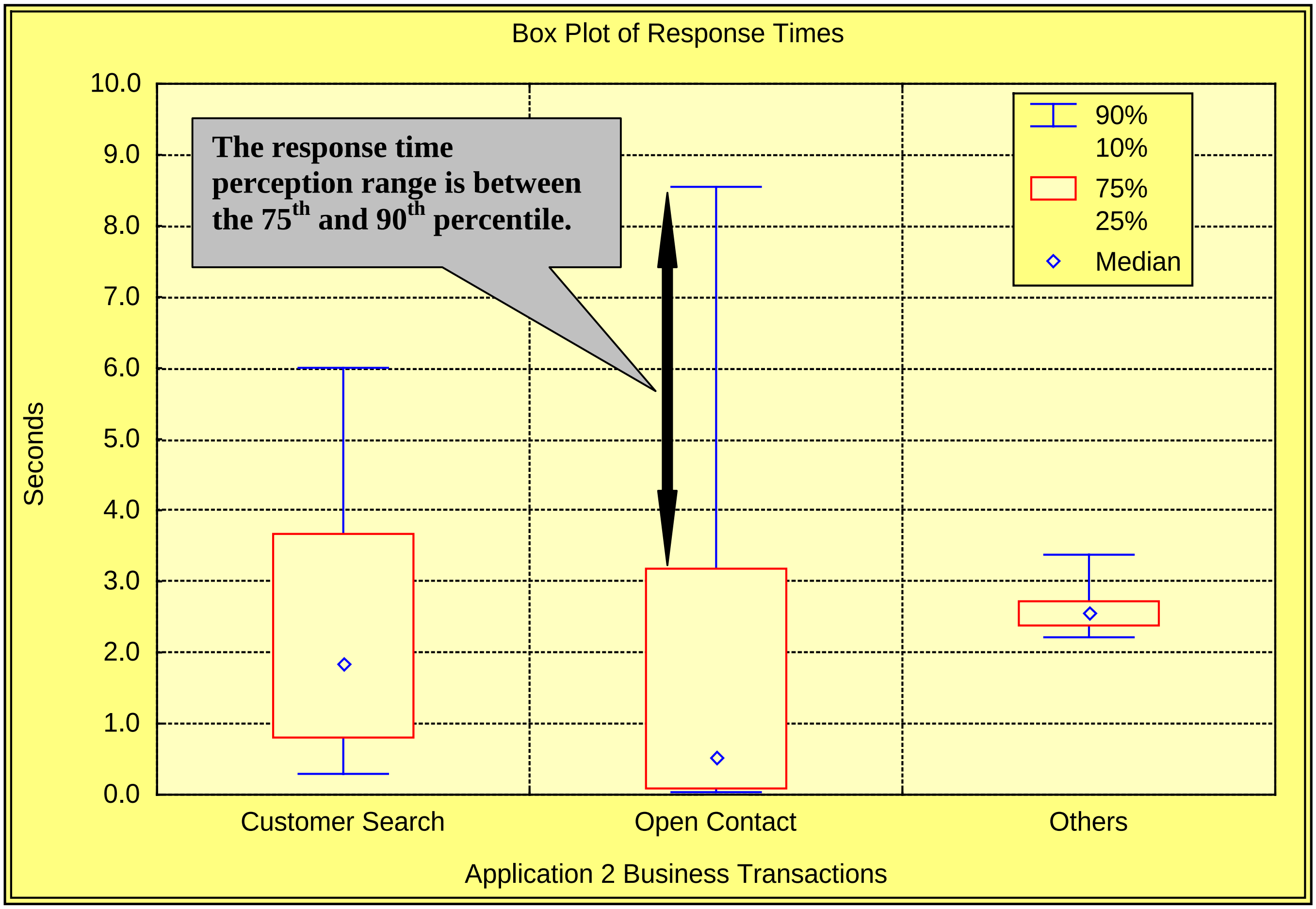


Figure 16 - Response Time Perception, shows the clients perception of response time is likely to be in a range between the 75th and 90th percentile response times. It is this perception of response time that will influence the clients view toward saying the response times from the "Others" transaction group provided the best overall responses.

Summary

This paper has presented an enterprise level approach to Proactive Performance Engineering (PPE). PPE is a decision support service provided for the purpose of proactively engineering quality performance into software applications and the system environments for Information Technology business solutions. Ideas and experiences of initiating a PPE service in a large Information Technology organization were discussed. A contrived case study was presented for demonstrating the methods and techniques applied to building an enterprise level model. The initial steps toward building an enterprise level model start with a spreadsheet model of the business volume metrics. The steps required to transition from a spreadsheet model to a fully functional client/server model in *Strategizer®* were demonstrated. Finally, a cursory validation of the framework provided in the *Strategizer®* case study model was discussed as well as ideas for presenting response times of the business transactions.

References

[BREYFOGLE III1992] Breyfogle III, Forrest W., *Statistical Methods For Testing, Development, and Manufacturing*. Published by John Wiley & Sons, Inc., 1992.

[CRYSTAL BALL®] Users Manual, Version 4.0, 1988-1996. Decisioneering, Inc.

[LOOSLEYDOUGLAS1998] Loosley, Chris & Douglas, Frank, *High-Performance Client/Server*. Published by John Wiley & Sons, Inc., 1998.

[SES] Scientific and Engineering Software, Inc.

[SMITH1990] Smith, Connie U., *Performance Engineering of Software Systems.* Addison-Wesley Publishing Company, Inc., 1990.

[TANNER1995] Tanner, Mike, *Practical Queueing Analysis*. IBM McGraw-Hill Series, 1995.

Trademarks