HUMAN-COMPUTER INTERACTION, 1993, Volume 8, pp. 237-309
Copyright © 1993, Lawrence Erlbaum Associates, Inc.

Project Ernestine: Validating a GOMS Analysis for Predicting and Explaining Real-World Task Performance

Wayne D. Gray
NYNEX Science & Technology Center

Bonnie E. John
Carnegie Mellon University

Michael E. Atwood
NYNEX Science & Technology Center

ABSTRACT

Project Ernestine served a pragmatic as well as a scientific goal: to compare the worktimes of telephone company toll and assistance operators on two different workstations and to validate a GOMS analysis for predicting and explaining real-world performance. Contrary to expectations, GOMS predicted and the data confirmed that performance with the proposed workstation was slower than with the current one. Pragmatically, this increase in performance time translates into a cost of almost $2 million a year to NYNEX. Scientifically, the GOMS models predicted performance with exceptional accuracy.

Authors' present addresses: Wayne D. Gray, Fordham University, Room 1008, 113 West 60th Street, New York, NY 10023. Email: gray@mary.fordham.edu; Bonnie E. John, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213. Email: Bonnie.john@centro.soar.cs.cmu; Michael E. Atwood, NYNEX Science & Technology, Inc., 500 Westchester Avenue, White Plains, NY 10604. Email: atwood@nynexst.com.

CONTENTS

The empirical data provided us with three interesting results: proof that the new workstation was slower than the old one, evidence that this difference was not constant but varied with call category, and (in a trial that spanned 4 months and collected data on 72,450 phone calls) proof that performance on the new workstation stabilized after the first month. The GOMS models predicted the first two results and explained all three.

In this article, we discuss the process and results of model building as well as the design and outcome of the field trial. We assess the accuracy of GOMS predictions and use the mechanisms of the models to explain the empirical results. Last, we demonstrate how the GOMS models can be used to guide the design of a new workstation and evaluate design decisions before they are implemented.

1. INTRODUCTION

"Design is where the action is," argued Newell and Card (1985, p. 214); to affect the field of human-computer interaction significantly, a theory or methodology must apply to design of a system, not merely to after-the-fact evaluation. Newell and Card argued further that to facilitate their application to design, psychological theories must quantitatively predict user performance from specifications of the task and of a proposed system, without relying on observations of human behavior with a working system or prototype.

One such theory is GOMS (Card, Moran, & Newewll, 1980a, 1980b, 1983), which analyzes behavior in terms of the user's Goals; the Operators available for accomplishing those goals; frequently used sequences of operators and subgoals (Methods); and, if there is more than one method to accomplish a goal, Selection rules to choose between them. GOMS is a family

of analysis techniques composed of goals, operators, methods, and selection rules; these components, however, are expressed in different ways, such as goal hierarchies (Card et al., 1980a, 1983), lists of operators (Card et al., 1980b), production systems (Bovair, Kieras, & Polson, 1990), working memory loads (Lerch, Mantei, & Olson, 1989), and schedule charts (John, 1988, 1990). The different expressions have different strengths and weaknesses and predict different measures of performance (e.g., operator sequences, performance time, learning, errors). Variants of GOMS have been used successfully to predict user behavior with computer systems in the laboratory (see Olson & Olson, 1990, for a review). In this study, we go outside the laboratory to assess a GOMS model's ability to quantitatively predict expert performance on a real-world task, to provide explanations for the predictions, and to direct future design activity.

1.1. Appropriateness of Project Ernestine for the Evaluation of GOMS

Project Ernestine was well suited for evaluating GOMS models for four reasons:

1. The important metric for the project is expert performance time, a strength of current GOMS modeling.
2. The task involves procedures easily modeled in GOMS.
3. A large-scale field study was run concurrently that provided data against which to validate the GOMS predictions.
4. An alternative (noncognitive, nonmodel) basis for prediction existed.

We explain each of these points in turn.

Expert Performance Time: A Valuable Real-World Metric

New England Telephone was considering replacing the workstations currently used by toll and assistance operators (TAOs) with new workstations. A major factor in making a buy/no-buy decision was how quickly the expected decrease in average worktime per call would offset the capital cost of making the purchase. Given the number of TAOs employed by NYNEX (the parent company of New England Telephone) and the number of calls it processes each year, a ballpark estimate is that an average decrease of 1 sec in worktime per call saves $3 million per year. Thus, differences in expert performance time attributable to workstation differences are economically important to NYNEX. Project Ernestine was initiated to provide a reliable estimate of these differences.

Predicting experts' time has always been a strength of GOMS modeling.

Such predictions are successful when the users are experts performing a routine cognitive skill and when they make few errors. These conditions were satisfied in Project Ernestine. A TAO handles hundreds of calls each day, and many stay at the job for years (even decades). Highly practiced, expert TAOs are easy to find. TAOs recognize each call situation and execute well-practiced methods, rather than engage in problem solving. Thus, TAOs are performing a routine cognitive skill. As for errors, the call-handling system is designed to preclude many types of errors (e.g., the workstation will not release a call unless all necessary information is entered), and experienced TAOs make few errors of any type.

The Nature of the TAOs' Task and Environment and Their Expression in GOMS

Successful GOMS models depend on the task having clearly identifiable goals, operators, methods, and (if necessary) selection rules. The task of a TAO has these characteristics.

A TAO is the person you get when you dial 0. A TAO's job is to assist a customer in completing calls and to record the correct billing. Among others, TAOs handle person-to-person calls, collect calls, calling-card calls, and calls billed to a third number. (TAOs do not handle directory assistance calls.)

TAOs are trained to answer three questions in the course of a call; these questions correspond to three goals in a GOMS analysis. For each call, a TAO must determine (a) who should pay for the call, (b) what billing rate to use, and (c) when the connection is complete enough to terminate interaction with the customer. For instance, for a person-to-person collect call, (a) the called party should pay for the call, (b) the person-to-person rate should be applied, and (c) the connection is not complete until the requested person is on the line, has accepted the charges, and the called number is eligible to accept collect calls (e.g., coin phones are not eligible).

To accomplish these goals, TAOs converse with the customer, key information into a workstation, and read information from the workstation screen. In some cases, they also write notes to themselves (primarily to remember callers' names). Thus, the GOMS operators for accomplishing the goals include listening, talking, reading, keying, writing, and the cognitive activities necessary to assimilate information and determine action. An additional complexity, beyond the sheer variety of activities, is that TAOs must perform many of these tasks simultaneously. Complex interactions among these activities dictate the call-completion time.

Many of these activities had been modeled successfully in GOMS research prior to Project Ernestine. Heuristics for modeling the perception of words on a display screen and for modeling typing had already been developed (John, 1988; John & Newell, 1989b; John, Rosenbloom, & Newell, 1985). In addition, a method for modeling parallel activities, CPM-GOMS, had been

added to the GOMS family of techniques (John, 1988; John & Newell, 1989b), and we use this variant of GOMS to model the parallel behavior displayed by the TAOs. To handle the variety of activities TAOs must perform, we made extensions to CPM-GOMS to model auditory perception, verbal responses, eye movements to get information from well-known positions on a CRT screen, and system response time. These extensions were made early in the project (John, 1990), and their derivation and general use is discussed in more detail elsewhere (John & Gray, 1992).

The TAOs accomplish their goals using a dedicated workstation, and that workstation influences the CPM-GOMS models. We evaluated two workstations, which we called *current* and *proposed.* The current workstation uses a 300-baud, character-oriented display and a keyboard on which keys are color coded and grouped by function. Information is presented on the display as it becomes available, often as alphanumeric codes, with the same category of information always presented at the same screen location (e.g., the calling number is always on the third line of the display). In contrast, the proposed workstation uses a high-resolution display operating at 1,200 baud, uses icons and windows, and in general is a good example of a graphical user interface whose designers paid attention to human-computer interaction issues. Similar care went into the keyboard, where an effort was made to minimize travel distance among the most frequent key sequences. In addition, the proposed workstation also reduces keystrokes by replacing common, two-key sequences with a single key.

These differences in the workstations affect the methods used to process a call as well as the duration of specific operators within a method (e.g., the duration of the horizontal movements to some keys changed between workstations). Expressing the differences between workstations in the models was a straightforward application of GOMS: No extension to the theory was required.

The Field Study: Data for Comparison to GOMS Predictions

For the purpose of validating GOMS models, the most important feature of Project Ernestine is that NYNEX conducted an empirical trial over a prolonged period of time using real TAOs handling live traffic (i.e., real calls from real customers). As we observed elsewhere (Atwood, Gray, & John, in press):

> Applying analytic modeling efforts to real-world settings is an enterprise laden with paradox. If models predict results that designers consider "intuitive," then the models are perceived to be of little value. On the other hand, if models predict results that are counter-intuitive, why, in the absence of empirical data, should they be believed? More importantly, why should an expensive empirical trial be conducted to validate

a counter-intuitive prediction of an analytic model? Understandably, opportunities to demonstrate the value of analytic modeling are rare.

Telephone companies have long known the value of field trials, especially when important economic decisions are to be made. Therefore, NYNEX scheduled the field trial independent of any modeling effort. This allowed us to do GOMS analyses as if the new workstation did not actually exist but was merely a proposed design and, after the analyses, to compare the predictions with independently collected empirical data.

Alternative Basis for Prediction

Empirical field trials are expensive and complex and are seldom conducted. The fact that NYNEX decided to conduct an empirical trial of a proposed workstation meant that NYNEX already was largely sold on the advantages of the proposed workstation. Such a trial would be expected to confirm expectations of worktime savings, as well as hardware and software reliability, while gaining in-house expertise in using and maintaining the new system.

NYNEX was expecting the proposed workstation to reduce average worktime because the proposed workstation eliminated one or more keystrokes for most calls, and it displayed a screenful of information in less time than the current workstation. A simple calculation based on these advantages predicted that the proposed workstation would reduce average worktime by almost 20% (see Section 4.3). Seeing that a simple, back-of-the-envelope calculation already predicted the difference in performance time, we were interested in whether the time and effort involved in building GOMS models would be justified by more accurate predictions.

1.2. The Structure of This Article

Our top-level goal for this article is to assess the validity of GOMS models for predicting and explaining performance on real-world tasks. To do this, we first discuss the model-building effort with particular attention to differences between using observational data for the current workstation and manufacturer-supplied specifications for the proposed workstation.

In Section 3, we discuss the field-trial per se and its empirical results. This discussion provides data against which to compare the quantitative predictions of the GOMS analysis and a baseline of information against which to argue the usefulness of GOMS.

In Section 4, we first evaluate the representativeness of the benchmark tasks on which we based the model-building effort (Section 4.1). We then compare our predictions to the field-trial data to determine the quantitative

accuracy of our models (Section 4.2) and see how these predictions stack up against an alternative, noncognitive basis for prediction (Section 4.3). We then go beyond the data and provide explanations for the results (Section 4.4).

Finally, in Section 5, we show what the models imply for the design of future workstations, demonstrating how such calculational models can be "tools for thought" (Newell & Card, 1985).

Before continuing, it is important to emphasize that the two major parts of Project Ernestine, the field trial and the GOMS analyses, were done separately and during the same time period. It is not the case that the GOMS models were built with knowledge of the empirical data. (Also, at the time of this writing, we have not observed a single TAO using the new workstation.) To better convey the parallelism of the trial and modeling, as well as to provide the reader with a spatial index to the various parts of this complex field study and modeling effort, we offer Figure 1.

2. BUILDING CPM-GOMS MODELS

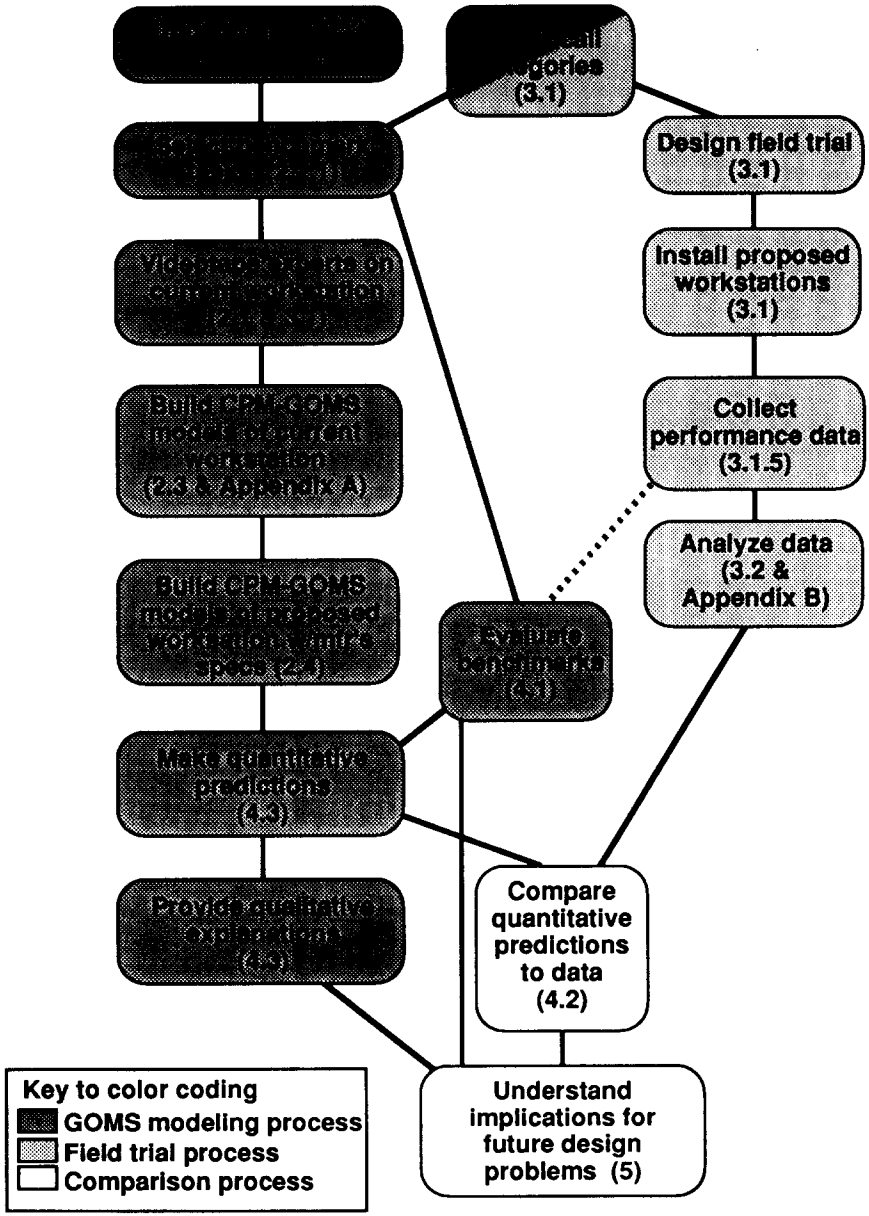
We begin this section with an analysis of the goal structure of the TAO's task. In that CPM-GOMS is a relatively new GOMS technique (John, 1990; John & Gray, 1992), we first present classic GOMS analyses of the TAO's task (Card et al., 1983) and use them to introduce the CPM-GOMS models actually used to predict TAO behavior. In Section 2.2, we discuss the issues and process of model building.

2.1. GOMS and CPM-GOMS

To show the continuity between classic GOMS and CPM-GOMS, we present analyses of the TAO's task at four levels: the unit-task level, the functional level, the activity level, and the Model Human Processor (MHP) level with CPM-GOMS. The unit-task and functional levels are directly analogous to the models of the same name in Card et al. (1983). The activity level is of finer granularity (somewhere between Card et al.'s argument and keystroke levels), and it distinguishes between different types of activities that models of text editing do not require. The CPM-GOMS analysis uses the critical path method technique explained in detail by John (1990; John & Gray, 1992). It uses operators at the level of the MHP's (Card et al., 1983) processor cycle times. The point of this comparison is that these analyses have the same roots, but they reveal different aspects of the user's performance and are useful for different aspects of design or evaluation.

The unit task for TAOs is the individual call. This corresponds to the individual edit discussed in Card et al.'s (1983) analysis of text editing. For example, consider the situation in which a person directly dials a number and

Figure 1. Flowchart of analyses in Project Ernestine (numbers refer to sections of the article).



has the TAO bill the call to his or her calling card. The dialog is sparse but typical:

Workstation:	[Beep]
TAO:	New England Telephone, may I help you?
Customer:	Operator, bill this to 412-555-1212-1234.
TAO:	Thank you.

The Unit-Task-Level Analysis

An analysis of the TAO's job, at the unit-task level, is shown in Figure 2. The single operator, HANDLE-CALL, has a fixed estimated duration. At this level of analysis, the time per unit task is constant and does not vary by call category. For the phone company, having an estimate of time per call of, say, 30 sec is very useful. In fact, such an estimate is combined with historical data on call volume to schedule TAOs by 15-min intervals for each day of the week, each week of the year. To predict differences in worktime by call category and to guide workstation design, however, we need a more detailed analysis.

The Functional-Level Analysis

The functional level provides more detail than the unit-task level. The subgoals at this level directly reflect the three questions the TAO must answer to handle the call: who pays, at what rate, and is the call complete? A functional analysis of our example call is shown in Figure 3.

In this functional-level analysis, all calls can be analyzed using only four operators: RECEIVE-INFORMATION, REQUEST-INFORMATION, ENTER-INFORMATION, and RELEASE-CALL. The TAO's task is so constrained that there are virtually no situations in which alternative methods arise, so selection rules do not play a role in these analyses.

Time estimates for the call are based on time estimates for each of these four operators. It is a simplifying assumption of GOMS models that each operator (at any level of analysis) has an estimated duration that may depend on inputs to that operator but not on the context of other operators in which it occurs. For instance, a typing operator could be defined to take the text to be typed as its input and a duration that is (approximately) a linear function of the number of characters. The functional-level operators we chose to define herein do not have such task-dependent inputs; each operator has just one time estimates despite the fact that it covers a variety of task situations. For instance, the RECEIVE-INFORMATION operator represents behavior ranging

Figure 2. **Unit-task-level GOMS analysis of example call.**

GOAL: HANDLE-CALLS
. GOAL: HANDLE-CALL . *repeat as calls arrive at workstation*

Figure 3. **Functional-level GOMS analysis of example call.**

GOAL: HANDLE-CALLS	
. GOAL: HANDLE-CALL	. *repeat as calls arrive at workstation*
.. GOAL: INITIATE-CALL	
... RECEIVE-INFORMATION	
... REQUEST-INFORMATION	
.. GOAL: ENTER-WHO-PAYS	
... REQUEST-INFORMATION	... *if additional information needed*
... RECEIVE-INFORMATION	
... ENTER-INFORMATION	
.. GOAL: ENTER-BILLING-RATE	
... REQUEST-INFORMATION	... *if additional information needed*
... RECEIVE-INFORMATION	
... ENTER-INFORMATION	
.. GOAL: COMPLETE-CALL	
... REQUEST-INFORMATION	... *if additional information needed*
... RECEIVE-INFORMATION	
... RELEASE-CALL	

from the fraction of a second necessary for the TAO to read the screen to determine if a call is from a COIN or NON-COIN phone to the several seconds needed to hear the customer say, "Operator, bill this to 412-555-1212-1234."

Although not all categories would have exactly the same pattern of functional-level operators, the variety of patterns will be small, and we are likely to end up with two or three different patterns to cover all 15 call categories. Indeed, the functional level has little to say about performance time differences between call categories or between workstations. To the contrary, for the TAO's job, the functional level emphasizes the commonality of functions across call categories. This level of analysis speaks to the design of workstations only to ensure that some procedure for accomplishing each subgoal is indeed provided by the workstation. Thus, this is the level of GOMS analysis best suited to guide the early stages of workstation design.

The Activity-Level Analysis

At the activity level, we begin to get a sense of the job being done, that is, how the TAO interacts with the customer and the workstation to handle the call. The operators from the functional level become subgoals composed of finer-grained operators. For example, the functional-level operator, RECEIVE-INFORMATION, becomes an activity-level goal, which is accomplished with one or more operators specific to the type of information being received (LISTEN-FOR-BEEP, READ-INFO-FROM-SCREEN, LISTEN-TO-CUSTOMER).

Figure 4 shows an activity-level analysis of the TAO's task. The conditionality on operators starts to be unwieldy at the activity level, so for illustrative purposes this figure shows only the sequence of goals and

Figure 4. **Activity-level GOMS analysis of the TAO's task.**

GOMS Model	Observed Activities
GOAL: HANDLE-CALLS	
. GOAL: HANDLE-CALL	
. . GOAL: INITIATE-CALL	
. . . GOAL: RECEIVE-INFORMATION	
. . . . LISTEN-FOR-BEEP	Workstation: Beep
. . . . READ-SCREEN(2)	Workstation: Displays information
. . . GOAL: REQUEST-INFORMATION	
. . . . GREET-CUSTOMER	TAO: "New England Telephone, may I help you?"
. . GOAL: ENTER-WHO-PAYS	
. . . GOAL: RECEIVE-INFORMATION	
. . . . LISTEN-TO-CUSTOMER	Customer: "Operator, bill this to 412-555-1212-1234."
. . . GOAL: ENTER-INFORMATION	
. . . . ENTER-COMMAND	TAO: Hit F1 key
. . . . ENTER-CALLING-CARD-NUMBER	TAO: Hit 14 numeric keys
. . GOAL: ENTER-BILLING-RATE	
. . . GOAL: RECEIVE-INFORMATION	
. . . . READ-SCREEN(1)	
. . . GOAL: ENTER-INFORMATION	
. . . . ENTER-COMMAND	TAO: Hit F2 key
. . GOAL: COMPLETE-CALL	
. . . GOAL: REQUEST-INFORMATION	
. . . . ENTER-COMMAND	TAO: Hit F3 key
. . . GOAL: RECEIVE-INFORMATION	
. . . . READ-SCREEN(3)	Workstation: Displays credit-card authorization
. . . GOAL: RELEASE-CALL	
. . . . THANK-CUSTOMER	TAO: "Thank you"
. . . . ENTER-COMMAND	TAO: Hit F4 key

operators for the specific credit-card call described before. In addition, the observable activities that the TAO engages in to handle this call are shown next to the operators that represent them in the model.

The activity level may be sufficient to identify the skills that TAOs need to have or need to be trained in to do the job (e.g., training in using standard greetings). It may also help to guide design questions. For example, the first **RECEIVE-INFORMATION** goal has two operators: **LISTEN-FOR-BEEP**, which alerts the TAO that a call is arriving, and **READ-SCREEN**, which provides information about the source of the call so the TAO can choose the appropriate greeting. Can one of these operators be eliminated? Can the beep both signal the arrival of a call and indicate the source of the call (e.g., with a small number of different-pitched tones)? At the activity level, differences between call categories would be mirrored in differences between the patterns of goals, subgoals, and operators.

Despite its uses, the activity level is not appropriate for predicting time differences either between workstations or between call categories. For workstations, the problem is obvious. At the activity level, the proposed and current workstations have exactly the same goal–subgoal–operator structure.[1] Hence, they would be predicted to have exactly the same duration.

For call categories, the situation is only slightly more subtle. Any operator is given the same estimated duration regardless of the variety of circumstances that it may encompass. For example, **LISTEN-TO-CUSTOMER** would have the same duration for the "Operator, bill this to 412–555–1212–1234" example as for "Operator, I want to make this collect to my Grandmother Stewart, who has been feeling ill lately, from her grandson Wayne" as for "Bill this to 1234." Hence, at the activity level, differences in the type and number of operators for each call category would tend to be overwhelmed by the range and variability of individual operators.

The activity level also highlights a problem with the sequential nature of the original GOMS models. As an illustration, suppose the durations of the observable operators (**LISTEN-TO-BEEP, GREET-CUSTOMER, ENTER-COMMAND, ENTER-CALLING-CARD-NUMBER,** and **THANK-CUSTOMER**) and system response time were set from a videotape of this sample call and an estimate of **READ-SCREEN** came from previous work reading short words from a CRT screen (John & Newell, 1989a, 1989b). Then the sum of these operators and system response times predicts that the call would take almost 17.85 sec (see Figure 5). In reality, this sample call takes 13 sec to complete.

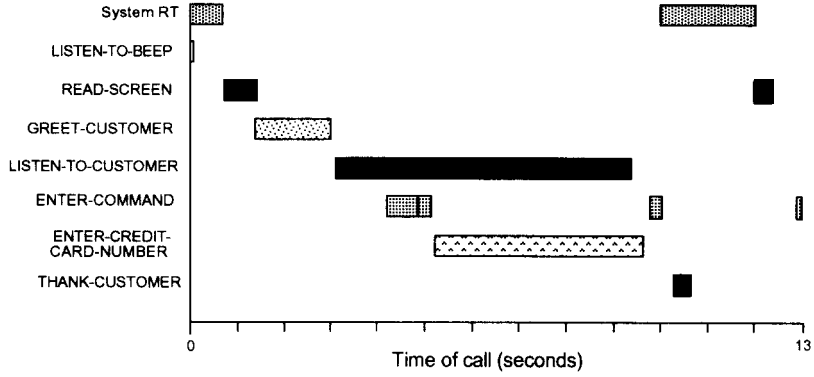
[1] Although this statement is true for the two workstations that were compared in Project Ernestine, it may not be true in general. For Project Ernestine, the proposed workstation did not change the nature of the job being performed. For situations in which a change in workstation was also accompanied by a change in the job, say from the use of a drafting table to the use of a CAD/CAM system, then analysis at the activity or functional level may be sufficient to bring out critical differences.

Figure 5. Activity-level prediction of task performance time (total predicted = 17,850 msec; total observed = 13,000 msec; percent error = 37%). Notes: We include observed system response time when it does not overlap with user behavior as per Card, Moran, and Newell (1983). Operators corresponding to observed behavior with more than one occurrence (only the **ENTER-COMMAND** operator) are assigned a duration equal to the average of all observed occurrences; operators with only one occurrence are assigned a duration identical to that occurrence (all other operators set from the videotape). Unobservable operators (only the **READ-SCREEN** operator) are assigned a duration based on prior research, as noted.

GOMS Model	Duration Estimates	Operator Source of Estimate
GOAL: HANDLE-CALLS		
. GOAL: HANDLE-CALL		
. . GOAL: INITIATE-CALL		
. . . GOAL: RECEIVE-INFORMATION		
. . . . LISTEN-T0-BEEP	100 msec	Videotaped call
(SYSTEM-RESPONSE-TIME)	730 msec	Videotaped call
. . . . READ-SCREEN(2)	340 msec	John and Newell (1987, 1989a)
. . . GOAL: REQUEST-INFORMATION		
. . . . GREET-CUSTOMER	1,570 msec	Videotaped call
. . GOAL: ENTER-WHO-PAYS		
. . . GOAL: RECEIVE-INFORMATION		
. . . . LISTEN-TO-CUSTOMER	6,280 msec	Videotaped call
. . . GOAL: ENTER-INFORMATION		
. . . . ENTER-COMMAND	320 msec	Videotaped call (average)
. . . . ENTER-CALLING-CARD-NUMBER	4,470 msec	Videotaped call
. . GOAL: ENTER-BILLING-RATE		
. . . GOAL: RECEIVE-INFORMATION		
. . . . READ-SCREEN(1)	340 msec	John and Newell (1987, 1989a)
. . . GOAL: ENTER-INFORMATION		
. . . . ENTER-COMMAND	320 msec	Videotaped call (average)

.. GOAL: COMPLETE CALL		
... GOAL: REQUEST-INFORMATION		
.... ENTER-COMMAND	320 msec	Videotaped call (average)
... GOAL: RECEIVE-INFORMATION (SYSTEM-RESPONSE-TIME)	2,000 msec	Videotaped call
.... READ-SCREEN(3)	340 msec	John and Newell (1987, 1989a)
... GOAL: RELEASE-CALL		
.... THANK-CUSTOMER	360 msec	Videotaped call
.... ENTER-COMMAND	320 msec	Videotaped call (average)

Figure 6. **Time line of activities in the sample call. Note: The durations of the observable operators are the actual durations in the videotape, not the averaged durations used in the GOMS analysis in Figure 5. The unobservable operator, READ-SCREEN, is shown with the duration estimated from the literature (340 msec) and is positioned just after the relevant information appears on the TAO's screen.**



The reason for this is obvious from a time line of actual events recorded in the videotape (Figure 6). The TAO is clearly performing activities concurrently, and the GOMS analysis does not capture this dominant feature of the task. The assumption of strictly sequenced operators substantially overpredicts performance time for this task, even when using measured values for most of the operator durations. Therefore, to make quantitative predictions of TAOs' performance requires a more powerful representation for parallel activities—that of the CPM-GOMS analysis technique.

The CPM-GOMS Analysis

The CPM-GOMS extension to classic GOMS is linked closely to the cognitive architecture underlying GOMS: the MHP (Card et al., 1983). The MHP has three processors: a cognitive processor, a perceptual processor, and a motor processor. In general, these processors work sequentially within themselves and in parallel with each other, subject to information-flow dependencies. To display parallel activities and information-flow dependencies and to calculate total task times, we use the critical path method, a common tool used in project management. Thus the CPM-GOMS analysis technique gets its name from expressing Cognitive, Perceptual, and Motor operators in a GOMS analysis using the Critical Path Method.

To model the TAOs' tasks, perceptual operators are divided into two categories: visual and auditory operators. Motor operators are divided into four categories: right-hand, left-hand, verbal, and eye-movement operators. Cognitive operators are not further divided into categories. These operators

are at a finer grain than the operators of the previously mentioned activity-level analysis; that is, the activity-level operators become CPM-level goals, with the MHP-level operators combining to accomplish these goals. The details of where MHP-level operators come from in general, estimates for their duration, and how they combine to accomplish activity-level goals are presented elsewhere (John, 1990; John & Gray, 1992), but we discuss building TAO task-specific models in more detail in the next section.

Other processors exist in this human–computer system (e.g., the call-switching processor, the databases of credit card numbers, and the workstation itself). The operators of these processors are represented by two categories of system response time: **workstation display time** and **other systems-rt** (where *rt* means response time). Although **system-rt** is not a true GOMS operator, we refer to it as an operator for simplicity of exposition.

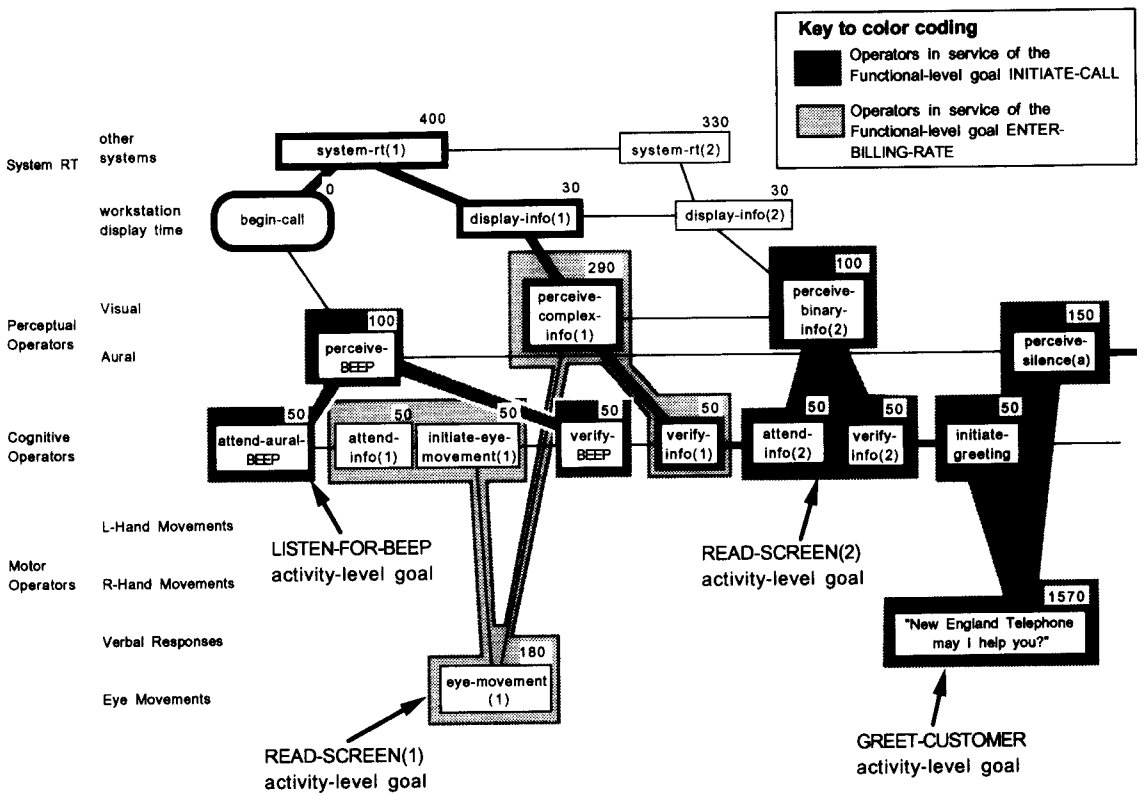
Only one operator in each category can be performed at a time; that is,operators are serial within category. However, they can be performed in parallel with operators in other categories.

Schedule Charts: A Notation for Representing Parallelism. In CPM-GOMS, the parallelism of the TAOs' task is represented in a *schedule chart* (Figure 7). Each MHP-level operator is represented as a box with a name centered inside it and an associated duration above the top-right corner (in msec). Lines connecting the boxes represent information-flow dependencies; that is, when a line joins two operators, the operator to the left produces information required by the operator to the right. For visual clarity, we place operators of the same category along a horizontal line.

The goal hierarchy of the classic GOMS analysis is not explicitly represented in the schedule chart, but it is implicit in the operators that are represented. For example, an activity-level operator **READ-SCREEN(1)** in Figure 4 represents the TAO reading the screen to find information about the billing rate (about halfway down the figure). This becomes an activity-level goal in the CPM-GOMS model, **READ-SCREEN(1)**, and is accomplished by five MHP-level operators explicitly represented in Figure 7 (backed by light gray shading): **attend-info(1)**, a cognitive operator that decides to look for this information; **initiate-eye-movement(1)**, a cognitive operator that initiates an eye movement to where that information will appear on the screen; **eye-movement(1)**, a motor operator that positions the eyes; **perceive-complex-info(1)**, a visual perception operator that takes in and comprehends the information when it is displayed; and finally **verify-info(1)**, another cognitive operator that confirms that the information is as expected (i.e., not something unexpected like a workstation failure producing an error message).

We represent the information-flow dependencies between these operators by drawing lines between them. There is a line from **attend-info(1)** to **initiate-eye-movement(1)** because the TAO must first decide to find this

Figure 7. Example CPM-GOMS schedule chart. To illustrate the higher level goal structure common to classic GOMS but not explicit in CPM-GOMS, for this sample the higher level goals are indicated by groups of operators and the background color of the groups (light vs. medium gray). Each group indicates a different activity-level goal; each background color indicates a different functional-level goal.



information before determining that an eye movement is necessary. There is a line from **initiate-eye-movement(2)** to **eye-movement(1)** because the eyes cannot move until a cognitive operator instructs them to do so. **Eye-movement(1)** is connected to **perceive-complex-info(1)** because the information cannot be perceived until the eyes are in the right place. Finally, there is a line between **perceive-complex-info(1)** and **verify-info(1)** because the information must be perceived and comprehended before a cognitive operator can verify that it is indeed the information that was expected. The rationale for these dependencies is based both on common sense and on the control structure of the MHP (Card et al., 1983).

In addition to the previously described MHP-level operators that represent the actions of the TAO, the actions of the workstation also play a part in this example. The information that the TAO needs to determine the billing rate is not displayed instantaneously at the beginning of the call. Rather, the other systems take 400 msec to deliver that information to the workstation, represented by **system-rt(1)**, and the workstation itself takes 30 msec to display that information, **display-info(1)**. The TAO cannot perceive the information until it is displayed on the screen, so there is an additional dependency line from **display-info(1)** to **perceive-complex-info(1)**.

This pattern of five MHP-level operators, linked to the display of information on the screen, implicitly represents that the **READ-SCREEN** activity-level goal is being served.

Altogether, Figure 7 shows four patterns of MHP-level operators that implicitly represent four activity-level goals: **LISTEN-FOR-BEEP**, **READ-SCREEN(1)**, **READ-SCREEN(2)**, and **GREET-CUSTOMER**. Note that the pattern of MHP-level operators that achieves similar goals is not identical. For example, in Figure 7, **READ-SCREEN(1)** requires an eye movement that is not needed for **READ-SCREEN(2)** because the TAO's eyes are already in the right place in the course of the task. Likewise, **READ-SCREEN(1)** has a **perceive-complex-info** MHP-level operator whereas **READ-SCREEN(2)** has a **perceive-binary-info** MHP-level operator because the information needed for billing, **display-info(1)**, is a complex code that must be comprehended whereas the information needed to initiate the call, **display-info(2)**, is simply whether any information is displayed in a particular place on the screen or whether it remains blank.

The patterns of operators in Figure 7 are backed in either light or medium gray. The background shading indicates that these operators accomplish activity-level goals that are in service of different functional-level goals found in Figures 4 and 5. The operators backed in light gray accomplish **READ-INFO(1)**, which is in service of **RECEIVE-INFORMATION**, which is, in turn, in service of **ENTER-BILLING-RATE**. The operators in medium gray accomplish **LISTEN-FOR-BEEP**, **READ-SCREEN(2)**, and **GREET-CUSTOMER**, which are in service of **RECEIVE-INFORMATION** and

REQUEST-INFORMATION, which serve the functional-level goal of **INITIATE-CALL**. Thus, the goal hierarchy of the classic GOMS analyses is implicitly represented in patterns of MHP-level operators in the CPM-GOMS schedule chart.

The Critical Path. An important concept in analyzing the total time for tasks involving the complex interaction of parallel activities is the *critical path.* The critical path is the sequence of operators that, because of their durations and dependency relationship to other operators, determines the total time of the task. In CPM-GOMS models, the critical path is indicated by a bold outline of the operators' boxes and bold dependency lines between them. The sum of the durations of the operators on the critical path is the total time for the task.

For example, before deciding on a particular greeting for this customer, the TAO perceives the first piece of information displayed on the screen (**perceive-complex-info(1)**; see Figure 7) and then the presence or absence of the second piece of information (**perceive-binary-info(2)**). The perception, comprehension, and verification of this information (**perceive-complex-info(1)** and **verify-info(1)**) plus turning attention to the presence of the second piece of information (**attend-info(2)**) take longer than the system response times to deliver and display information (**system-rt(2)** and **display-info(2)**). Therefore, the critical path goes through the human activity rather than through the system response times, and the path is lined in bold. In this case, the systems' response times are said to have *slack time,* are not on the critical path, and are not lined in bold. Human activity that has slack time has also been observed in videotapes of TAOs handling calls. For example, TAOs move to specific function keys then hover over them while waiting for the customer or the workstation to give information that will dictate which key to press.

Comparison of the GOMS and CPM-GOMS Analyses

The classic GOMS models of the TAO's task (Figures 2, 3, and 4) and the CPM-GOMS model (Figure 7 and Appendix A) look very different. Although CPM-GOMS is an extension to classic GOMS, its graphic representation of operators and dependencies can obscure the theoretical roots common to both analysis techniques as well as the theoretical differences between them.

Both techniques start with a decomposition of the task into a hierarchy of goals and subgoals that determines which operators are performed. However, because durations are only assigned to operators, not to the manipulation of goals (Card et al., 1983), only operators contribute to the total task duration. CPM-GOMS schedule charts are a representation of task duration and thus do not include any explicit representation of goals. The fact that goals are explicitly represented in the classic GOMS goal-operator lists but only

implicitly represented in the CPM-GOMS schedule charts is a superficial difference between the two techniques.

However, there are two important theoretical differences between classic GOMS and CPM-GOMS: parallelism of operators and relaxing the hierarchical goal structure. CPM-GOMS does not assume sequential application of operators to accomplish a goal as does classic GOMS. The MHP-level operators are performed in parallel with those of other categories, subject to resource limitations and task-specific information-flow dependencies described before.

In addition, CPM-GOMS does not enforce the stack discipline of classic GOMS, in which only one goal can be active at a time. In CPM-GOMS, different goals can exist concurrently. In Figure 7, for example, the TAO reads the first two pieces of information that are displayed on the screen at the very beginning of the call (**perceive-complex-info(1)** and **perceive-binary-info(2)**, respectively). As it happens, the second piece of information displayed is the information required to choose the appropriate greeting in service of the first functional-level goal in the classic GOMS analysis, **INITIATE-CALL** (Figure 3). The first piece of information displayed is part of the information required to determine the appropriate rate in service of a later functional-level goal, **ENTER-BILLING-RATE** (Figure 3). The **ENTER-BILLING-RATE** goal cannot be completed until the customer supplies more information later in the call, but the TAO assimilates this information opportunistically as it appears on the screen. Thus, this CPM-GOMS model has two functional-level goals active at the same time: **INITIATE-CALL** (represented by the three operator patterns backed by medium gray in Figure 7) and **ENTER-BILLING-RATE** (backed by light gray).

A second example occurs at a lower level with activity-level goals in Figure 7. Concurrent activity-level goals are evident in how the MHP-level operators that accomplish the **LISTEN-FOR-BEEP** activity-level goal (backed by medium gray) are interleaved with the operators that accomplish the **READ-SCREEN(1)** goal (backed by light gray). These situations would have been impossible to represent in classic GOMS, but they more accurately reflect observed behavior in the TAO's task.

The four previously presented analyses are a basis for understanding the TAO's task in general and its representation in CPM-GOMS. The next section explains the specific model-building procedure we used to model TAOs' performance with the current workstation and to predict their performance with the proposed workstation.

2.2. The Benchmark Method

The *benchmark method,* which compares two or more systems constructed to accomplish the same task, is a well-established technique in engineering

disciplines. The method involves choosing a set of *benchmarks* that typify the tasks, running the systems on those benchmarks, and (for each benchmark) comparing performance across systems.

Benchmarks are invariant across the systems to be compared. For example, to compare database architectures, Benigni, Yao, and Hevner (1984, 1985) held constant the size and content of the test data, the computer's workload, and the specific queries put to the database. Similarly, Roberts and Moran (1983) developed benchmarks for text editors that held constant the location and amount of text to be inserted, deleted, moved, and so on. Likewise, Hillelsohn (1984) used the development of 30 frames of a benchmark lesson to compare five authoring systems for computer-based training. For Project Ernestine, each benchmark represents a different call category. Within a benchmark, we keep constant all activities and times except those explicitly dictated by changes in workstation design or procedures. For example, we hold constant the exact variant of call within the call category, the TAO's dialog, customer's dialog, and all system responses external to the workstation (e.g., database checks on calling-card numbers).

Benchmarks are commonly used to compare performance data for existing systems. However, they also can be used with simulations of systems when the systems do not exist or are costly (or dangerous) to run with the benchmark tasks. We chose to use benchmarks with simulations in the form of CPM-GOMS models because we wished to test the validity of using CPM-GOMS at the specification stage of a design, before a running system is built.

To perform the CPM-GOMS analysis with the benchmark method, first we selected appropriate benchmarks. We then determined the CPM-GOMS operators necessary to perform these benchmark tasks on the current workstation, dependencies between them, and estimates of their durations. Finally, we built CPM-GOMS models for all the benchmark tasks using the proposed workstation and made quantitative predictions of performance.

Twenty call categories were selected because of either their high frequency or their particular interest to NYNEX Operator Services. Benchmark tasks were developed for each of these 20 call categories. For a variety of reasons, which are fully explained in Appendix B, however, five of the call categories were dropped from the empirical analysis and are not considered further in this article.

Calls within a category vary considerably (Appendix B, Figure B-3 shows that the coefficients of variance for calls in the empirical trial on the current workstation ranged from 0.37 to 1.05). This variation is primarily due to variations in customer conversation. For instance, in a credit-card call, a customer might give her entire 14-digit calling-card number, or, if the call is to the customer's home phone, she may give the TAO only her four-digit personal identification number. Additionally, some customers leave out

information, requiring the TAO to ask them for it. Some customer conversations require this prompting, some do not, causing much of the variance.

In the face of such expected variation, we strove to script a single benchmark call for each category so that the set of benchmarks would be representative of the types of calls TAOs would actually handle. With the help of NYNEX Operator Services personnel, we varied the customer conversation in the benchmark scripts to reflect major differences in customer behavior such as those described before.

We then used the benchmark scripts to place staged calls to an expert TAO using a current workstation. The TAO would put the call through, and, when required by the call category, a confederate played the role of the called party (using another phone within the trial office). We videotaped eight TAOs handling the staged calls for approximately 1 hr each; the TAOs knew the calls were staged and that they were being videotaped. For each call category, we selected one instance on the videotapes that both followed the script and contained all the information we needed (e.g., the complete conversation between the TAO and the "customer," a good recording of the display screen, and good recording of the TAO's hand movements). These single instances of the staged calls became the final benchmarks for workstation comparison.

2.3. Observation-Based CPM-GOMS Models for the Current Workstation

The next step in constructing CPM-GOMS models to predict TAO performance is to build observation-based models of the benchmark tasks performed on the current workstation. We used the videotaped behavior on the current workstation to determined MHP-level operators necessary to perform these tasks, dependencies between them, and estimates of their durations.

First, we transcribed the videotape for the 15 selected benchmark calls. Each transcript included the actual start and stop times for each verbal communication, for each keystroke, and for each change on the workstation display (to the accuracy of a video frame, 32 msec).

After the transcript was made, we created a CPM-GOMS schedule chart reflecting the procedures observed in the videotapes, observable operators, and (using the classic GOMS models as a guide) our best guess for both the unobservable operators and the dependencies (details can be found in John, 1990, and John & Gray, 1992). Because we were modeling a single benchmark for each call, there was no opportunity for us to observe different methods for accomplishing a call or to infer selection rules if different methods existed. However, because the TAO's task is so constrained that often only a single correct method corresponds to a particular script, this

simplification seems reasonable. Note that, in a few of the videotapes we analyzed, we observed inefficient keying strategies, typically extra keystrokes. NYNEX Operator Services personnel tell us that some of these inefficient strategies are holdovers from older, now obsolete, procedures. Others may reflect a misunderstanding of procedures on the part of the videotaped TAOs, whereas others may simply be slips. We chose to model the inefficiencies as observed, rather than model the most efficient procedures possible on the current workstations. Because all of the videotaped TAOs were experts with at least 2 years of experience, we felt that our observations were representative of the actual behavior of expert TAOs on the current workstations.

We then made estimates of the operator durations. For observed operators, we used the actual durations, which we call *benchmark durations*. For the unobserved operators, we used *normative estimates* set from the literature. Figure 8 contains a list of all the types of operators used to model the TAOs' performance of the benchmark tasks on the current workstation, their estimated durations, and the source of those estimates.

Benchmark Durations

The durations for all pausing and speaking, both by the TAOs and the customers, are set by the benchmarks and taken directly from the videotapes. These include complex auditory perceptual operators, such as the real-time perception and comprehension of the customer's phrase "make this collect," as well as the duration for the TAO's motor operator to say "New England Telephone, may I help you?" The duration of hand movements for pressing keys on the current workstations is also set from the videotapes, as are all the system response times for the current workstations.

Normative Estimates

Normative estimates are obtained from the literature and are off-the-shelf estimates of how long an average person requires to perform a particular operator. We used nine normative operators. The four cognitive operators are assumed to be of equal duration, 50 msec (John & Newell, 1989a, 1989b). The motor operator that makes an eye movement to a known screen location is assumed to be 180 msec (this is the Card et al., 1983, estimate of 230 msec for an eye movement subdivided into a cognitive operator, 50 msec, to initiate the movement and the motor action itself, 180 msec). Binary visual perception operators, used when the TAO must detect only the presence or absence of a visual signal, are assumed to take 100 msec (minimal perceptual operator in the MHP; Card et al., 1983). For example, in Figure 7, the perception of info(2) is a binary perception; the TAO need only detect that a code appears in that spot on the screen, not what the code actually says (because it always

Figure 8. **Operator types and durations used for CPM-GOMS. CO = cognitive operator; MO = motor operator; PO = perceptual operator; SRT = system response time.**

Operator	Current Workstation		Proposed Workstation	
	Type	Duration	Type	Duration
CO: **attend-visual** <info>	Normative	50 msec	Normative	50 msec
CO: **attend-aural** <info>	Normative	50 msec	Normative	50 msec
CO: **initiate;** <motor response: keystroke, speech, eye movement>	Normative	50 msec	Normative	50 msec
CO: **verify-** <info>	Normative	50 msec	Normative	50 msec
MO: <speech>	Benchmark	As measured	Benchmark	As measured
MO: **horizontal-** <to key>	Benchmark	As measured	Predicted	100 msec
MO: **home-** <from lap to keyboard>	Benchmark	As measured	Benchmark	As measured
MO: **down-** <keystroke>	Benchmark	As measured	Predicted	90 msec
MO: **up-** <keystroke>	Benchmark	As measured	Predicted	100 msec
MO: **eye-movement**	Normative	180 msec	Normative	180 msec
PO: **perceive-auditory** <speech>	Benchmark	As measured	Benchmark	As measured
PO: **perceive-auditory silence**	Normative	300 msec	Normative	300 msec
PO: **perceive-BEEP**	Normative	100 msec	Normative	100 msec
PO: **perceive-visual-binary** <info>	Normative	100 msec	Normative	100 msec
PO: **perceive-visual-complex** <info>	Normative	290 msec	Normative	290 msec
SRT: <all SRTs including workstation display time and other system>	Benchmark	As measured	Predicted	Current ± Estimated

says the same thing).[2] In contrast, perceiving **info(2)** in Figure 7 is a complex visual perception because the TAO must perceive and comprehend the semantics of the code displayed, as well as the presence of the code, to get sufficient information to continue with the call. The complex visual perceptions required of the TAO are all of small words, alphanumeric codes, or numbers, and they are assumed to take 290 msec because they are of similar character to the small-word recognition tasks used to estimate that duration (this is the John & Newell, 1989a, 1989b, estimate of 340 msec for the perception and encoding of a short word subdivided into a 290-msec perceptual operator and a 50-msec cognitive operator to verify expectations). Binary auditory perceptual operators, such as detecting the "beep" that signals an incoming call, are also set at 100 msec (minimal perceptual operator of the MHP; Card et al., 1983). The perception of an auditory silence that signals turn taking in conversation is estimated at 300 msec (this is the 400-msec mean interspeaker pause found by Norwine & Murphy, 1938, subdivided into a 300-msec perceive-silence operator followed by a 50-msec cognitive operator to verify the silence and a 50-msec cognitive operator to initiate the spoken response).

Identifying Impossible Times and Computing the Critical Path

The first CPM-GOMS schedule chart included the actual durations as well as the actual start and end times of each observable operator and normative durations of the unobservable operators. This often redundant and overconstraining information caused the project management software (MacProject™) to highlight inconsistencies in the schedule charts. Such impossible times signaled problems with the initial model. Things that could be wrong at this point included our understanding of the call-handling procedures, our choice of operator grain size, the number of operators, or the dependencies between operators.

Impossible times led us to refine our understanding of the current workstation by reanalyzing the videotapes and/or by discussions with managers in NYNEX Operator Services. We revised the models based on our increased understanding of the task. This iterative process continued until most impossible times were eliminated. Some impossible times resulted from our placement heuristics for unobservable operators and could not reasonably be removed; these were on the order of tens of milliseconds and were considered a necessary consequence of the approximate nature of engineering models.

[2] Binary visual perceptions are followed by a cognitive operator to verify that the perception is what was expected (i.e., that the perception was of a code being displayed on the screen, not a sudden glare from someone turning on a nearby lamp or another visual event not related to the task).

When we were satisfied that the cognitive task analysis in the CPM-GOMS models was reasonably correct, we removed all start and end times from the schedule charts. The project management software then calculated the critical path based only on the durations of the operators and their dependencies, not on the observed temporal position of actual behavior. The resulting schedule charts show the critical paths in bold and produce predictions of the length of each benchmark call. A schedule chart of a full calling-card call is shown in Appendix A.

Comparing Current Models to Videotapes

We evaluated the predictions of the current models against the times observed in the videotapes from which they were built. Because the models were based on the videotaped calls, we expected the differences between the two to be small. These expectations were supported by the data.

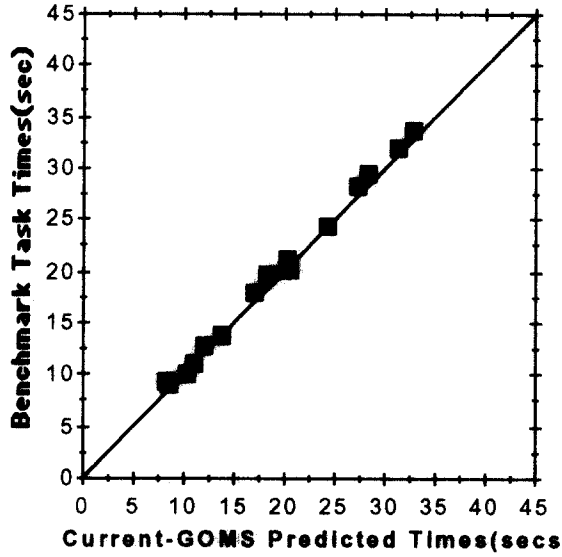
For each call category, the CPM-GOMS model underpredicted worktime by an average of six tenths of a second (3%). This finding suggests that there is some component of the videotaped call that is consistently missing from the model. However, we do not interpret this underprediction as bad news. We did not expect our models to be perfect. Because decomposition models tend to underpredict, we are very pleased that they are close to the videotaped worktimes. More important than the 0.6-sec difference is the fact that this difference is quite constant across call categories. The models are consistent in their underprediction, so that if, for example, they predict that cc01 (where *cc* = call category) will be slower than cc02, this is likely to be the case. This conclusion is supported by a significant correlation (r^2 = .996; see Figure 9).

2.4. Specification-Based CPM-GOMS Models of the Proposed Workstation

In contrast to modeling the current workstation based on videotaped calls, we modeled the proposed workstation without ever observing TAOs using that workstation. Our primary goal in this part of Project Ernestine was to validate the use of CPM-GOMS as an evaluation tool at the specification stage of design, in the absence of a working system. Thus, we treated the proposed workstation as if it did not yet exist.

For each call category, we modeled the proposed workstation as if this particular customer (in the videotaped call) called with these particular requests and were handled by this particular TAO using the proposed workstation. Therefore, things related to the customer-TAO dialog (e.g., wording, pauses, and duration of phrases) are the same for the models of the proposed workstation as for the models of the current workstation. Likewise, things related to individual TAO performance (e.g., typing speed) are the

Figure 9. **Compares the benchmark task calls with the times predicted by the current CPM-GOMS models. Note that the 45° diagonal illustrates where the points would fall if the predictions were perfect.**



same for both sets of models. Things specific to the workstation do vary: layout of the proposed display, keyboard arrangement, manufacturer-specified procedures, and manufacturer-supplied estimates of system response time.

The inefficiencies observed in the benchmark videotapes and modeled for the current workstation were not modeled for the proposed workstation. Rather, we followed the manufacturer-supplied procedures, only deviating if a New England Telephone practice would have resulted in faster predicted worktimes and if the deviation captured a component of the nonworkstation, system performance with which our TAOs were already familiar (e.g., taking advantage of the speed difference between a database check and call outpulsing[3]). Therefore, for the proposed workstation, our default policy was to use the procedures appearing in the manufacturer's manuals. Because we expected the specified procedures to be more efficient than the actual procedures, this decision should have biased the results (to an unknown

[3] This keypress strategy is very efficient. The call cannot be outpulsed until after the database check is completed. If the calling-card number is invalid, then, regardless of the TAO's keypress, the call will not go out and the TAO will have to converse with the customer. Pressing the key when the redisplay begins is the most efficient strategy for valid calling-card numbers. For invalid numbers, this strategy is neutral; that is, it is no faster or slower than waiting for the results of the database check to display.

degree) toward predicting shorter worktimes for the proposed workstation than what might actually be observed.

Just as in the models of the current workstation, durations of unobservable operators were set to normative estimates from the literature. Durations of benchmark-related operators (e.g., durations of words, customer pauses, and the time to hit numeric keys) were set from the videotapes. In addition, the models of performance on the proposed workstations used predicted durations for (a) the hand motions to function keys and (b) the system response times. Figure 8 shows the operators used in the proposed workstation models and the source of their durations.

The motor component of each keystroke in a CPM-GOMS model is composed of a horizontal movement to the key, a downstroke and an upstroke. Because the function keys on the proposed workstation had similar mechanical characteristics (i.e., size, shape, and feel) as those on the current workstation, we used the actual times from the benchmark videotapes for the downstroke and upstroke for function keys that had exact counterparts on the current workstation. For those proposed keystrokes for which there were no current counterparts, we used a predicted estimate of 90 msec for the downstroke and 100 msec for the upstroke (these estimates are the average downstroke and upstroke times found in the benchmark videotapes).

The proposed keyboard had a more ergonomic arrangement of function keys, with more frequently used keys much closer together, and closer to the numeric keypad, than the current keyboard. In the new arrangement, the function keys were very similar to the numeric keypad both in size and distance between keys. Fitts's law (Fitts, 1954) tells us that size and distance are the two determining factors in estimating horizontal movement; therefore, we used the average of the observed time of horizontal movement between numeric keypresses on the videotapes, 100 msec, as the estimated duration of horizontal movement between function keypresses in the proposed models. This estimate is substantially less than most of the horizontal movements to function keys observed in the videotapes for the current workstation.

The manufacturer supplied us with estimates of the differences in system response time between the current and proposed systems due to differences in the systems' hardware. We estimated system response time for the proposed workstation by adding or subtracting these estimates from the times observed in the benchmark videotapes of the current workstation.

2.5. Summary of CPM-GOMS Model Building

The modeling process resulted in 30 CPM-GOMS models in schedule chart form. Fifteen of these are the CPM-GOMS models for the benchmark

tasks as executed on the current workstation, and 15 are the CPM-GOMS models for the benchmark tasks as executed on the proposed workstation. Each schedule chart gives a prediction of the length of its benchmark call, which is an estimate of the length of that type of call as it occurs in the field. (NYNEX considers the absolute time in seconds to be proprietary knowledge; therefore, we cannot report the detailed quantitative predictions of these models. We do, however, report summary statistics in Section 4.) These predictions of the length of benchmark calls can be combined with information about the frequency of call categories to predict the differences in average worktime between the two workstations. We work through the quantitative predictions of the models and compare them to empirical field data in Section 4.

3. THE FIELD TRIAL AND DATA

The field trial was conducted in a working telephone company office, with experienced TAOs, with an unobtrusive data-collection technique. The data are very rich and can be used to address many issues. In this report, we use the data to highlight three issues most important for validating GOMS. First, are there reliable differences in TAO worktime between the two workstations? This difference also has practical significance, as each second of worktime is typically cited as being worth $3 million per year in operating costs. Second, if there are differences between workstations, are these due to workstation design per se or are they affected by the TAOs' unfamiliarity with a new device? That is, do the differences diminish over the 4 months of the trial? Third, are these differences constant or do they interact with the type of call? If they are constant, then there may exist some basic design or technological flaw that could be simply fixed. If they interact with call category, then there may be as many fixes required as there are call categories. Understanding how to fix any given call category would require an in-depth analysis of the human-human and human-computer interactions for that call category.

3.1. Field-Trial Methodology

Overview of the Trial Design. There were 24 TAOs on each workstation. Training for the proposed workstation group was conducted before the trial started and followed standard New England Telephone procedures. During the 4-month trial, all normal phone company procedures and management practices were followed. No new procedures or practices were introduced. Except for the new workstation, everything about their work remained the same; both groups worked their normal shifts, with their normal days off, throughout the trial. Data collection involved tapping into a pre-existing database and required no experimenter intervention or on-site presence.

Trial Size. The New England Telephone office used in the study employs about 100 TAOs and handles traffic in a major metropolitan area. For purposes of the study, 12 of approximately 50 current workstations were removed and replaced by 12 proposed workstations.

TAO Selection. All participants had at least 2 years of job experience, were scheduled to work continuously for the next 6 months, and had a "satisfactory" or "outstanding" job appraisal. The proposed TAOs were selected from a list of 54 who had volunteered to use the new workstation. Twenty-four were chosen based on performance and schedule matching as well as management and union policies (see Atwood et al., in press). Each proposed TAO was matched with a control, or current TAO, on average worktime per call and schedule (including shift worked and days off).

The rest of the office continued using the current workstation. The 24 current TAOs matched to the proposed TAOs did not realize that they were part of the study and were not treated any differently from the other TAOs using the current workstation. (Note that all references to the current TAOs refer only to this matched, control group.)

We matched TAOs on their average worktime based on performance measures routinely collected by office managers for the 6 months prior to the trial. The difference in average worktime for the two groups was 0.09 sec,[4] with the proposed group very slightly, but not significantly, slower than the current group ($F < 1$).

Training on the Proposed Workstation. Training for the TAOs using the proposed workstations consisted of a 2½-day course conducted on-site by New England Telephone managers. In accordance with standard New England Telephone practice, most training was conducted on the proposed workstation with trainees handling increasingly complex types of live traffic. The manufacturer reviewed the training materials and found no discrepancy between what they felt were the proper training procedures and what New England Telephone included. Managers were trained before serving as instructors to the TAOs. The number of students in a class varied from two to four (typically two).

Data Collection. Information on every call handled by New England Telephone is routinely maintained in a database. New England Telephone uses a software product, OSCAS™ (supplied by BellCore) to produce a variety of reports from this database. Accessing this database allowed us to unobtrusively extract the data we required.

[4] NYNEX considers the absolute time in seconds to be proprietary knowledge. In discussing the results, we use the difference in seconds and the percent difference. However, all statistics were calculated using actual times.

An OSCAS call category is defined by call type (e.g., collect or calling card), dial type (e.g., coin or noncoin), and class (station or person). With the help of Bellcore OSCAS Software Development, we obtained a monthly report for every month of the trial. This report sampled 1 call out of every 10 and listed its duration in seconds, labeled by its category and the TAO handling the call.

For our purposes, the call and dial-type categories were overly broad, so we filtered the data to define the categories more sharply. For example, some OSCAS categories included calls that were handled by more than one TAO. In this case, we restricted our call set to single-TAO calls.

Call Category Baseline Frequency and Selection. Our study included 20 call categories selected because of either their high frequency or their particular interest to NYNEX Operator Services. Based on a pretrial OSCAS report that summarized calls for all of New England Telephone for 1 month, these categories accounted for 88% of all completed calls.

3.2. Field-Trial Results

In-depth and detailed analyses of the empirical data were conducted. Those readers not interested in these details are advised to skip ahead to the Summary of Data Analyses section. The analyses reported here are centered about two analyses of variance (ANOVAs). The first ANOVA looked at all 4 months of trial data and, among other things, found a significant Group × Month interaction. We were able to isolate this effect as being due to learning by the proposed group in the first month of the trial. For purposes of the field trial, per se, this first ANOVA yielded all the information NYNEX needed to know. For modeling purposes, however, we wanted to compare all of our model predictions against expert performance, that is, against the empirical estimates of worktimes derived from the data once learning had occurred and performance had reached its asymptote. Hence, as a conservative procedure, we redid the ANOVA using just the data against which we planned to compare our CPM-GOMS predictions.

ANOVA of the Trial Data

Over the 4 months of the trial, we collected data on 78,240 calls from 20 call categories performed by 48 TAOs. For a variety of reasons, which are fully explained in Appendix B, the following analyses are based on 72,390 calls from 15 call categories with 23 TAOs per group.

A mixed-design ANOVA was conducted with Groups (Current vs. Proposed) as an independent factor and Call Categories (15), Months (4), and TAOs (23) as repeated factors. The ANOVA summary table is shown in Figure 10.

Figure 10. **Analysis of variance: Group × Call Category × Month.**

Source	*df*	Sum of Squares	Mean Square	*F*	*p*
Group	1	1,131.25	1,131.25	6.11	.0174
Subject (Group)	44	8,152.75	185.29		
Call Category	14	198,412.66	14,172.33	441.48	.0001
Call Category × Group	14	617.02	44.07	1.37	.1608
Call Category × Subject (Group)	616	19,774.93	32.10		
Month	3	578.81	192.94	10.52	.0001
Month × Group	3	145.14	48.38	2.64	.0523
Month × Subject (Group)	132	2,421.46	18.34		
Call Category × Month	42	1,645.21	39.17	2.59	.0001
Call Category × Month × Group	42	663.43	15.80	1.05	.3935
Call Category × Month × Subject (Group)	1848	27,938.37	15.12		

For all 15 call categories and 4 months of the trial, the median worktime for the proposed group was 106.0% that of the current group; that is, for the average call on the average month, the proposed workstation required 6% (1.3 sec) more time than did the current workstation.

This counterintuitive result, new technology being slower than old technology, is both practically and statistically significant (see Figure 10), $F(1, 44) = 6.11$.[5] It indicates that switching to the proposed workstation would increase worktimes and, as a result, incur higher annual operating costs than the current workstation. (Note, too, that these yearly operating costs are in addition to such one-time transition expenses as initial equipment, installation, and initial training.)

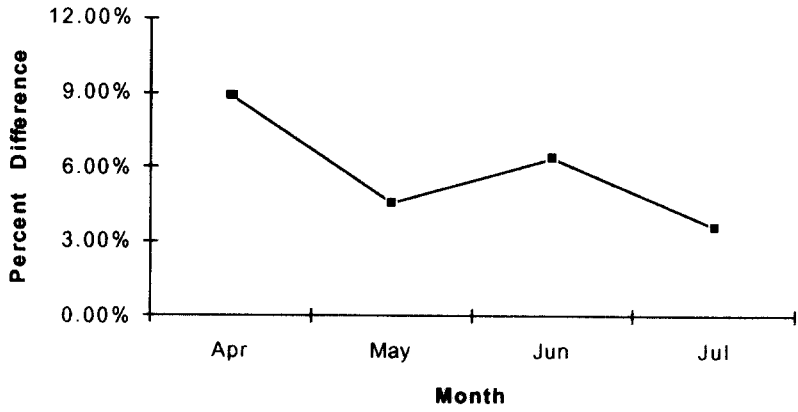
The effect of call category is also significant, $F(14, 644) = 441.48$, confirming what has always been obvious to New England Telephone; some call categories are longer or shorter than others. A bit more surprising[6] is that the effect of call category did not interact with group, $F(14, 616) = 1.37$, $p = .16$. This absence of an interaction suggests that the disadvantage of the proposed workstation is relatively constant across call categories.

The main effect of month is also significant, $F(3, 132) = 10.52$. This is another confirmation of something New England Telephone has always known. Seasonal differences in the call mix cause monthly fluctuations in

[5] Note that, unless stated otherwise, $p < .05$ is the significance level used throughout this study.

[6] We reported such a significant interaction in our CHI '92 paper (Gray, John, & Atwood, 1992). Those preliminary analyses were based on a less conservative procedure for identifying outliers (see Appendix B) and on a two-factor ANOVA (Group × Call Category) rather than the three-factor ANOVA (Group × Call Category × Month) used here.

Figure 11. **Percent by which the proposed is slower than the current workstation by month: [(Current × Proposed)/Current] × −100.**



average worktime. Of more importance for us is the nearly significant Month × Group interaction, $F(3, 132) = 2.64$, $p = .0523$. As can be seen in Figure 11, this is an ordinal interaction; that is, the relative rankings of the two groups do not change with respect to month. The proposed group is always slower. Although a significant interaction is usually seen as invalidating conclusions regarding the main effect, in the case of ordinal interactions this does not hold (Keppel, 1973, p. 204). Hence, although the Month × Group interaction is an important phenomena in its own right, it does not invalidate the conclusion that the proposed group is significantly slower than the current group.

Although we are confident that the proposed group was slower than the current group throughout the trial, the marginally significant Month × Group interaction raises the possibility that the proposed group was getting faster and, if the trial had continued long enough, would have been faster than the current group. This important possibility is discussed and further analyzed next.

Figure 10 also shows that the effect of call category interacts with month, $F(42, 1848) = 2.59$. This interaction most likely reflects seasonal variations in the types of calls and callers and as such is not relevant to the discussion of workstation differences.[7] The Call Category × Month × Group interaction is not significant, $F(42, 1848) = 1.05$.

[7] For example, the phone company believes that business callers are more efficient and faster than callers in general. To the extent that the ratio of business callers to, say, vacationers varies by month for a given call category, the expectation is that worktime will vary.

Is Learning Occurring? Comparisons by Month

The difference between groups varies from 8.83% in April to 4.53% in May to 6.35% in June to 3.56% in July. The marginal interaction suggests that some learning was occurring during the trial. Was learning localized to the first month of the trial, or was it continuous throughout? If the latter is true, then if the trial had continued long enough eventually the proposed group might have become faster than the current one.

To answer this question, we partitioned the sum of squares for the interaction (Keppel, 1973). As shown in Figure 10, the interaction reflects a sum of squares of 145.14 divided by 3 *df*. Following Keppel, the 3 *df* permit us to partition this sum of squares into 3 independent, single degree-of-freedom, comparisons of the interaction.

We looked first at the Group × Month interaction for April versus the rest of the trial. With a sum of squares of 110.25, 1 *df*, and a mean square of 110.25, the comparison was significant, $F(1, 132) = 6.01$, $p = .016$. The current versus proposed difference for April was significantly larger than the difference for the rest of the months.

Technically, we could partition the Group × Month interaction into two more independent comparisons. However, because the previous comparison used up 110.25 out of 145.14 of the available sum of squares, the remaining sum of squares (145.14 − 110.25 = 34.89) would not be significant even in the unlikely event that it all was located in one of the two comparisons. That comparison would yield the following, $F(1, 132) = 1.90$, $p = .17$. We conclude that the proposed group shows learning during April but that the worktimes from May onward represent stable differences between the two workstations.

This conclusion introduces a new complication. Our CPM-GOMS models are models of expert performance. We do not expect them to predict novice performance or changes in performance with increasing expertise. We do expect them to predict the stable expert performance that we find in the May, June, and July data. Hence, in comparing the CPM-GOMS predictions with the empirical data, we look at worktimes based on the stable 3 months of performance. April's data were dropped completely. However, before discussing the models' predictions, as a conservative procedure, we present a second ANOVA identical to the previous one but based on the stable 3 months of empirical data.

Reanalysis of the Trial Using Data From May, June, and July

Figure 12 shows the results from the second ANOVA. Except for Months, which now has three levels (May, June, and July) rather than four, all other factors and data are identical to those used in the first ANOVA, and the results are almost identical as well.

Figure 12. **Analysis of variance: Group × Call Category × Month, where Month is May, June, and July only (April was dropped).**

Source	*df*	Sum of Squares	Mean Square	*F*	*p*
Group	1	570.11	570.11	4.15	.0476
Subject (Group)	44	6040.00	137.27		
Call Category	14	142,970.45	10,212.18	423.52	.0001
Call Category × Group	14	320.56	22.90	0.95	.5046
Call Category × Subject (Group)	616	14,853.34	24.11		
Month	2	161.34	80.67	5.61	.0051
Month × Group	2	34.85	17.42	1.21	.3028
Month × Subject (Group)	88	1,266.02	14.39		
Call Category × Month	28	877.63	31.34	2.40	.0001
Call Category × Month × Group	28	333.42	11.908	.91	.5976
Call Category × Month × Subject (Group)	1232	16,078.58	13.05		

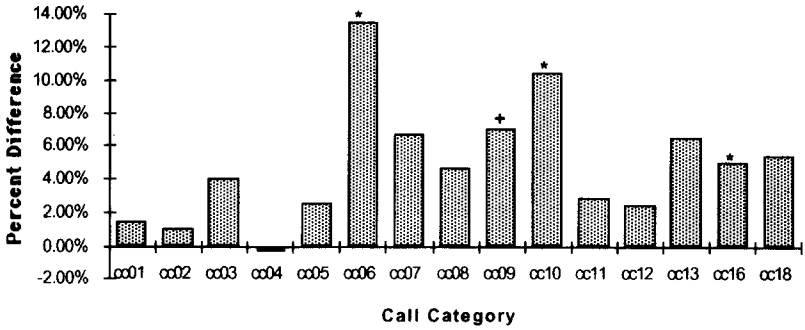
As Figure 12 shows, except for the Month × Group interaction, all comparisons that were significant in the first ANOVA remained significant. All comparisons that were not significant remained insignificant. Most important, the main effect of group was significant, showing that the proposed workstation is significantly slower than the current one.

This analysis shows that, after asymptotic performance is reached, worktime for the proposed workstation is 104.8% that of the current workstation. Thus, for the average call on the average month, the proposed workstation required 4.8% (1.05 sec) more time than does the current workstation.

The only comparison whose significance changed was that of the Month × Group interaction. Although this comparison aroused considerable concern in the first ANOVA, it is not significant here, $F(2, 88) = 1.21$, $p = .3028$. We conclude that at asymptotic performance the proposed workstation is slower than the current one.

The Call Category × Group interaction did not reach significance. However, as shown in Figure 13, for some of the call categories the worktime difference between workstations was small (for cc04, the proposed workstation was 0.1% faster!) whereas for others it was quite large (cc06 was 13.5% slower). Indeed, failure to find a significant Call Category × Group interaction may be a case where the overall ANOVA is obscuring a pragmatically important finding. The ANOVA weights each call category equally. In fact, the call categories vary dramatically in their frequency of occurrence. As Figure B-1 shows (see Appendix B) for the most frequent call category, cc01, we recorded data on 17,817 calls whereas for the least

Figure 13. **Percent difference in Workstations × Call Category for the 3-month data (* indicates a significant difference between current and proposed workstations; + indicates a marginally significant difference).**



frequent call category, cc18, we recorded data on 1,251 calls. This 14:1 ratio in the trial data reflects the relative occurrence of these call categories in the NYNEX world. For NYNEX, the importance (or lack thereof) of workstation differences by individual call category depends greatly on the frequency of occurrence of the individual call category.

Hence, the absence of a significant interaction cannot be interpreted as implying that the disadvantage of the proposed workstation is constant across all call categories. Because of the practical significance of this issue to NYNEX, we proceeded with our planned comparisons of current versus proposed groups by individual call category. A series of unpaired, two-tailed t tests yielded significant differences for cc06, $t(44) = 2.00$; cc10, $t(44) = 2.06$; and cc16, $t(44) = 1.99$. A marginally significant effect was found for cc09, $t(44) = 2.0$, $p = .06$. In all cases, the current group was faster than the proposed one.

Weighting Call Categories by Their Frequency of Occurrence

Earlier we stated that the proposed workstation was 4.8% (1.05 sec) slower than the current workstation. In that analysis, we weighted each call category equally, regardless of frequency of occurrence. However, because some call categories are much more frequent than others, estimates of operating costs must consider these differences. When worktime differences are weighted by call category frequency, the proposed workstation is 3.4% (0.65 sec) slower than the current one. At $3 million per second, these calculations represent a difference in annual cost of almost $2 million.

Summary of Data Analyses

These analyses of the empirical data have yielded three conclusions. First, performance with the proposed workstation is slower than with the current

workstation. Second, this difference is due to the workstation rather than to learning to use a new machine. Although performance on the proposed workstation improves during the first month of the trial, it remains slower than the current workstation, and this difference does not diminish over the last 3 months. Third, whatever causes the difference in worktime is not a simple constant. That is, for some call categories there is a slight advantage (0.1% for cc04) whereas for others the worktime disadvantage is quite large (13.5% for cc06).

The following discussion compares these findings with the predictions of the models built in Section 2.

4. COMPARING THE CPM-GOMS MODELS TO THE DATA

The CPM-GOMS models of the 15 benchmark tasks provide both quantitative and qualitative predictions about real-world performance on the current and proposed workstations. In this section, after evaluating the representativeness of our selected benchmarks (Section 4.1), we examine the quantitative predictions of the models and compare them to the data in the field trial (Section 4.2). In Section 4.3, we compare the CPM-GOMS predictions with those derived from a "reasonable alternative" calculation. We then look for qualitative explanations for the quantitative results (Section 4.4).

4.1. Evaluating Benchmark Tasks

An important step in using the benchmark method for comparing systems is to confirm that the benchmarks are indeed representative of the tasks for which the systems will be used. This is particularly important for our benchmarks because we used only a single script, performed by a single TAO, for each call category, and we did not deliberately design the scripts to approximate the median-duration call of each call type. For each call category, if this single instance was far from an average duration for that category, it would produce an abnormally long or short call duration for both workstations, with unknown effects on the predicted relative efficiency of the two workstations. Therefore, we compared performance time on the scripted calls to calls handled by TAOs on the current workstation during the trial. (Recall that NYNEX routinely collects average worktime by call category, so the ability to compare the benchmark times to real times was not dependent on the performance of a field trial.)

In terms of absolute worktime predictions, the average percent difference (see Figure 14) between trial times and videotape times over all call categories was 8.24%, with the benchmarks averaging slightly faster times than the real calls. When we weight the benchmark calls by the observed call frequencies,

Figure 14. **Percent difference between the 3-month current workstation trial data and the benchmark calls: (Current - Benchmark)/Current.**

Call Category	Percent Differences
cc01	-6.67%
cc02	-32.44%
cc03	20.45%
cc04	18.17%
cc05	7.07%
cc06	49.46%
cc07	-8.30%
cc08	45.71%
cc09	29.67%
cc10	-74.79%
cc11	18.88%
cc12	41.27%
cc13	8.30%
cc16	8.68%
cc18	-1.81%
Mean difference	8.24%

the prediction of worktime is even better: 2.08% less than the observed worktime. Simple linear regression shows that the correlation of call category times between the videotapes and trial data on the current workstation is significant ($r^2 = .70$; see Figure 15).

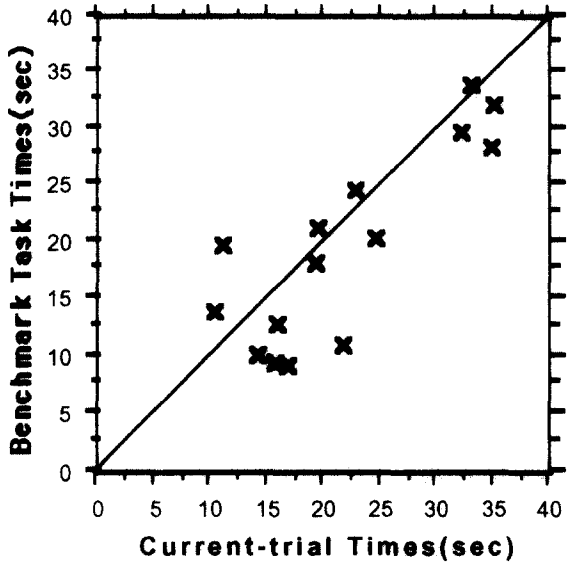
We also computed the standard score (z score) for each call category. The standard score expresses the difference between the videotape and trial means as a function of the variability of the trial data (difference/*SD*),[8] so that a z score between +1.00 and -1.00 indicates that the benchmark is within 1 *SD* of the mean of the trial calls. The standard scores showed that the benchmarks for 14 of the 15 call categories are within 1 *SD*.[9] These analyses support the conclusion that the set of benchmarks are representative of the calls TAOs handle in the real world.

Although in the aggregate, the set of benchmarks is representative of the set

[8] Because we were dealing with such a large amount of data (78,240 calls), we typically left the individual call data on the mainframe and downloaded the summary data that we used in our statistics. For part of the last month of the trial, however, we did download data on every call for every TAO for every call category. This database resulted in 16,059 calls: 8,125 for the current workstation and 7,934 for the proposed. Calls per call category ranged from 154 to 1,795. We use this database to supply estimates of worktime variability within call category.

[9] The benchmark used for cc10 had the customer giving much more information than the TAO needed to complete the call. The TAO politely waited until the customer had finished talking before answering her and completing the call. This procedure satisfied the constraint of serving the customer politely, but it produced a call time almost twice what was found in the trial in which, presumably, the average customer was less verbose.

Figure 15. **Compares the times of the benchmark tasks calls with current trial data. Note that the 45° diagonal illustrates where the points would fall if the benchmarks predicted the trial data perfectly.**



of real calls a TAO handles, the individual benchmarks varied widely in how well they matched their respective call types. The percent differences between times for the individual benchmarks and the trial data ranged from −74.79% to +49.46%, and 6 of the 15 call categories differed from the trial data by more than 20%. In retrospect, it might have been better to collect baseline performance data on the current workstation prior to modeling and use it to help design benchmark scripts. Depending on the observed distribution of a call category, a single benchmark could be designed to be close to the median, or several benchmarks could be designed to reflect a nonnormal distribution (e.g., a bimodal distribution resulting from some major difference like the 14-digit vs. four-digit calling-card scenarios described in Section 2.2).

4.2. Predicting Duration: Quantitative Validity

The CPM-GOMS models predicted durations for each of the benchmark calls for each of the workstations. Here we examine these predictions and compare them to the data from the field trial. First, we look at the relative difference between workstations. Does CPM-GOMS predict the 0.65-sec difference between current and proposed workstations when weighted by call frequency? Second, we look at the absolute worktimes for each workstation. How well do the models predict the absolute time for each of the two

workstations? Third, for each workstation we look at the absolute difference between prediction and field data for each of the 15 call categories. Finally, for each call category, we look at the relative difference between workstations.

Predicting the Difference Between the Two Workstations

When each model is weighted by the frequency of occurrence of its call category, CPM-GOMS predicts that the proposed workstation will be 0.63 sec slower than the current workstation. For comparison, when the empirical data are weighted by the frequency of call occurrence, the proposed workstation is 0.65 seconds slower than the current.

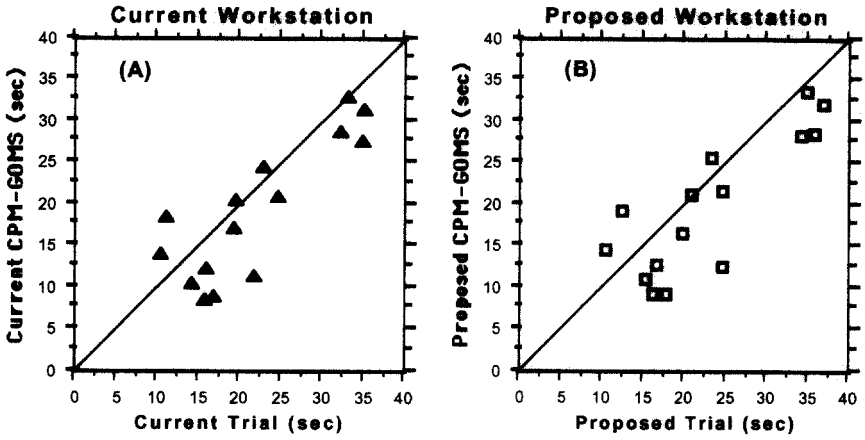
This overall prediction is the one that is most important to NYNEX. Pragmatically, at $3 million in operating costs per second of average worktime per year, the ability to predict performance on the mixture of calls that NYNEX TAOs handle is the most prized prediction. An analytic model of a proposed workstation, which was built without direct observation and based only on knowledge of the task and specifications obtained from the manufacturer, predicted, a priori, performance on that workstation in a real-world setting. Contrary to everyones expectation that the new technology would be significantly faster than the old technology, the models predicted a small difference favoring the old technology. This small difference in worktime meant that the proposed workstation would cost NYNEX an estimated $2 million per year more than the current workstation in operating costs. The CPM-GOMS models predicted the overall outcome of the trial with remarkable accuracy.

Predicting the Absolute Worktime for Each Workstation

For the current workstation, the CPM-GOMS models, when weighted by call category frequency, underpredict the trial data by an average of 4.35%. This underprediction is continued by the models of the proposed workstation, with these models predicting a weighted worktime 4.31% faster than the trial data. These weighted predictions are well within the 20% error limit that previous work (John & Newell, 1989a) has argued is the useful range of an engineering model.

Because these underpredictions are very consistent at about 4%, the relative prediction of the two sets of CPM-GOMS models (0.63 sec predicted vs. 0.65 sec found in the empirical data) is more accurate than the absolute predictions themselves. It is possible that this underprediction represents factors that are consistently missed by CPM-GOMS modeling. If further research also shows this consistent underprediction, then future analysts might consider adding a 4% adjustment to make more accurate absolute predictions of performance time.

Figure 16. **Regression scatterplots for call categories in seconds. (A) For the current workstation, comparison of CPM-GOMS predictions with the trial data. (B) Comparison for the proposed workstation. Note that the 45° diagonal illustrates where the points would fall if the predictions were perfect.**



Predicting Absolute Worktime by Call Category

Across call categories, the average percent difference between the CPM-GOMS models and the observed calls was 11.30% for the current workstation and 11.87% for the proposed workstation. The regression scatterplots of predicted versus actual times (Figure 16) show that the correlation between the CPM-GOMS predictions and the trial data was significant for the current workstation ($r^2 = .71$) and for the proposed workstation ($r^2 = .69$). For each workstation and call category, the standard z scores show that for 14 of the 15 call categories the CPM-GOMS prediction is within 1 *SD* of the trial mean for both current and proposed. These data support the conclusion that the CPM-GOMS models predict the trial data in the aggregate.

As with the benchmark tasks, the individual predictions of worktime per call category were less accurate. The percent difference per call category for the current workstation ranged from −63% to +49%, with eight call categories more than 20% away from their observed times (see Figure 17). Likewise, the percent difference for the proposed workstation ranged from −54% to +49%, with the same eight call categories being more than 20% away from the observed times.

The scatterplots in Figure 16 are extremely similar to the scatterplot of the measured times for the benchmark tasks versus trial data shown in Figure 15. This is not at all surprising for the current workstation because, in that the models predicted the benchmarks very well, their prediction of the trial data

Figure 17. **For the current and proposed workstations, the percent difference between CPM-GOMS predictions and the 3-month trial data: (Trial − GOMS)/Trial.**

Call Category	Current	Proposed
cc01	−6.06%	−9.57%
cc02	−32.92%	−36.96%
cc03	24.71%	23.27%
cc04	16.46%	12.16%
cc05	12.14%	16.60%
cc06	48.63%	49.17%
cc07	−2.61%	−0.38%
cc08	49.24%	49.23%
cc09	29.04%	29.00%
cc10	−62.86%	−54.45%
cc11	21.73%	20.63%
cc12	47.76%	44.39%
cc13	11.93%	17.94%
cc16	11.10%	12.98%
cc18	1.26%	4.10%
Mean difference	11.30%	11.87%

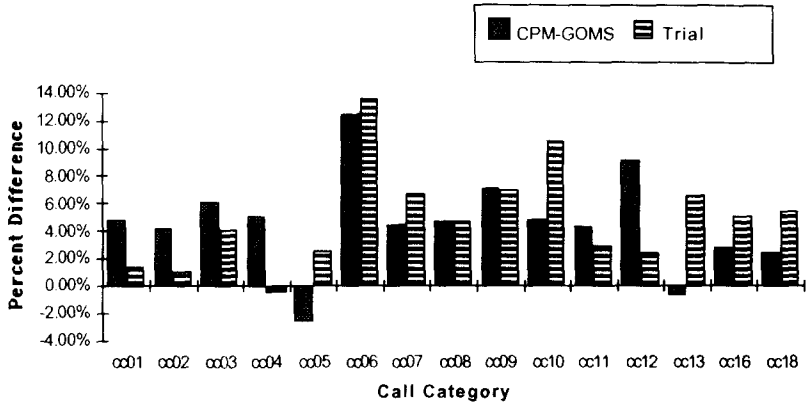
should be similar to the benchmarks' representativeness. Likewise, because there is much more variability between call categories than between workstations, the scatterplot for the proposed workstation (Figure 16B) also looks similar to that of the benchmarks (Figure 15).

These general results, that overall prediction of worktime (both weighted by call frequency and unweighted) is very good whereas the individual predictions of call category is not as good, is a statistical fact of life. If the individual predictions vary more or less randomly around the actual call times, some being too short and some being too long, aggregate measures will involve some canceling out of these predictions. Because the aggregate measures are of primary importance to NYNEX, this fluctuation at the level of individual call types is interesting to examine but not too important to the results of the modeling effort.

Predicting the Workstation Difference by Call Category

As can be seen in Figure 18, CPM-GOMS predicted the direction of the difference for all but three call categories (cc04, cc05, and cc13). We view the predictions as somewhat akin to EPA mileage ratings. Few drivers get the exact mileage predicted by EPA. However, the ratings are meaningful in that they tend to predict the direction of the differences between cars and the general size of those differences.

Figure 18. **Predicted percent difference in Workstations × Call Category.**



Summary: Quantitative Validity of CPM-GOMS Models

The CPM-GOMS models of benchmark calls predicted that the proposed workstation would be 0.63 sec slower than the current whereas the field trial found a real difference of 0.65 sec. For each workstation, the 15 CPM-GOMS models predicted worktimes that correlated highly with the empirical data (r^2 = .71 for current and .69 for proposed). Additionally, for 12 of the 15 call categories, the models predicted the direction of the current versus proposed difference.

In the next section, we compare the quantitative validity of the CPM-GOMS predictions with those derived from a reasonable alternative calculation. In Section 4.4, we use the CPM-GOMS models to do something that the data do not do (viz., provide a qualitative explanation of the differences between the current and proposed workstations).

4.3. Value Added of CPM-GOMS Models

As mentioned in the introduction, a simple, seemingly reasonable calculation can be done to predict worktime differences between the current and proposed workstations without cognitive modeling. Such a calculation was made before Project Ernestine, which raised NYNEX's initial expectations of improved performance with the proposed workstation and justified the expense of the field trial. Here we work through such a calculation and compare its accuracy to the CPM-GOMS predictions to evaluate the value added of cognitive modeling in the form of CPM-GOMS.

The benchmark tests can be used to make seemingly reasonable predictions of worktime differences between the current and proposed workstations without cognitive modeling. The proposed workstations displays a screenful of information faster than the current workstation and changes the keying

procedure to eliminate keystrokes for several call categories. For each call category, we work through these changes to predict overall differences in worktime.

From Card et al. (1983, Figure 9.1, p. 264) we get an estimate of 280 msec per keystroke for an average, 40 words/minute, nonsecretary typist. For each call category, this time was substracted for each keystroke that the manufacturer's procedures eliminated. Four keystrokes were eliminated from one benchmark call; two keystrokes from two calls; one keystroke from each of seven calls; zero keystrokes from four calls; and one keystroke was added to one call.

The manufacturer estimated that the proposed workstation would be 880 msec faster than the current workstation to display a screenful of information. We subtracted this estimate from every benchmark call.

By this benchmark-based, noncognitive procedure, we would predict an average advantage for the proposed workstation of 5.2%. When call categories are weighted by their frequency of occurrence, the predicted advantage becomes 18.6% (4.1 sec), for an estimated savings in annual operating costs of $12.2 million.

In contrast, the CPM-GOMS models predicted, and the field trial confirmed, that the proposed workstation would actually be about 3% slower than the current workstation. Thus, the seemingly reasonable calculation based on the benchmarks and the manufacturer's procedures and response-time estimates is wrong in both magnitude and sign. It is important to remember that the noncognitive prediction is more than just a straw man. Large-scale empirical trials such as Project Ernestine are expensive to conduct. Expectations based on such a calculation led NYNEX to commit to the time and expense required to conduct an empirical trial.

Why were the CPM-GOMS predictions so much more accurate than the noncognitive predictions? Two reasons are apparent: (a) Building CPM-GOMS models requires that the analyst understand the details of information flow between the workstation and the TAO, which were overlooked by the noncognitive predictions, and (b) CPM-GOMS models incorporate the complex effects of parallel activities.

For example, the noncognitive model assumed that each time a screenful of information was displayed, the proposed workstation's faster system response time would reduce the time of the call. However, the more detailed analysis required to build CPM-GOMS models revealed that the TAO does not have to see the entire screen to initiate the greeting (i.e., just the first line is needed). Hence, comparisons of how fast the two workstations display an entire screen of information are largely irrelevant. Likewise, the noncognitive model assumes that every keystroke contributes to the length of the call. However, CPM-GOMS shows that removing a keystroke only speeds the task if that keystroke is on the critical path.

Thus, CPM-GOMS disciplines the analyst to incorporate the right level of

detail to evaluate such tasks and correctly calculates the effects of parallel activities to produce accurate quantitative predictions. A noncognitive approach based on benchmarks and design changes alone does not work as well. In addition to producing more accurate quantitative predictions, CPM-GOMS models can provide qualitative explanations for the quantitative results (see next section) and can also be used as a tool in workstation design (Section 5). Clearly, CPM-GOMS adds value over noncognitive predictions.

4.4 Explaining Differences: Qualitative Validity

Beyond predicting performance time, the CPM-GOMS models provide explanations for their predictions and, thus, explanations for the empirical data. Here, we inspect the models to find what causes their differences in worktime and why these difference are not constant but vary with call category. We also pursue the intriguing absence of evidence for learning over the 4-month trial. The CPM-GOMS models take this simple null finding and, like Sherlock Holmes with "the dog that had not barked" (Doyle, 1892/1986, p. 475), find much importance in the absence of an expected event.

Why Do the Workstations Differ?

Despite its improved technology and ergonomically superior design, performance with the proposed workstation was slower than with the current workstation. A high-order look at the critical paths shows the task to be dominated by conversation and system response time. Seldom is the TAO's interaction with the workstation on the critical path. This pattern is so strong that it was found in our initial model of just one call category (Gray, John, Lawrence, Stuart, & Atwood, 1989) and so consistent that we declared it confirmed (Gray, John, Stuart, Lawrence, & Atwood, 1990) after modeling five call categories. Thus, the top-order prediction of the CPM-GOMS analyses is that the design of the workstation should have little, if any, effect on the length of calls.

We can look at the details of the models to understand why the proposed workstation is actually slower than the current workstation. The workstations differ in their keyboard layout, screen layout, keying procedures, and system response time, each of which may affect call duration.

Keyboard Layout. Compared to the current workstation, the keys necessary for the TAO's task are more closely grouped on the proposed workstation, with the most common keys clustered around the numeric keypad. Although this arrangement tends to reduce keying time, most keystrokes are

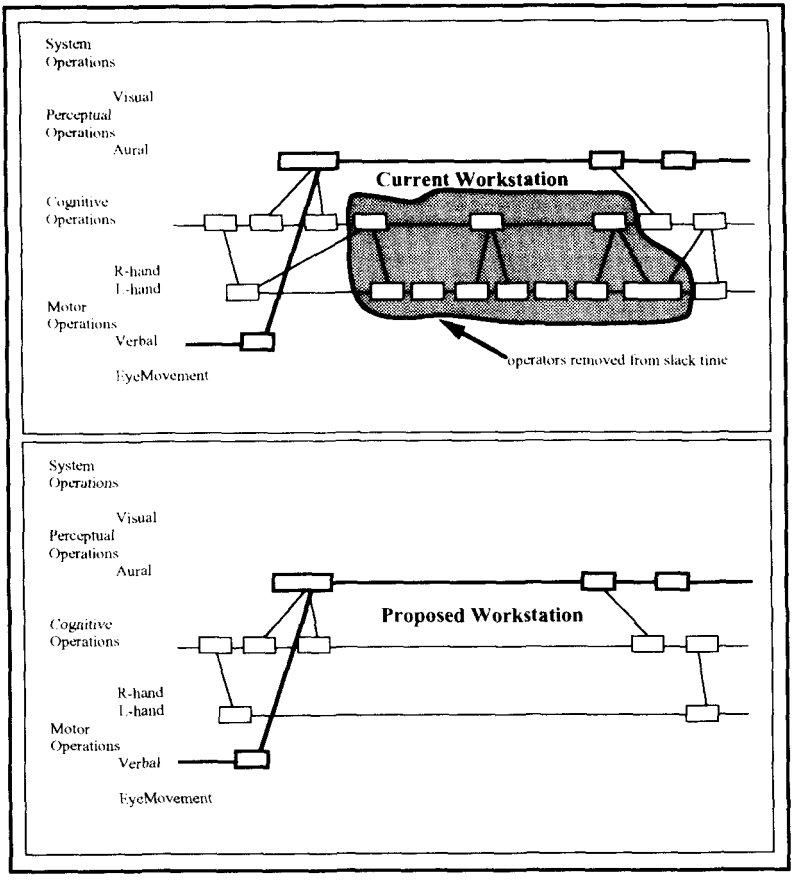
not on the critical path, so this advantage disappears into slack time for most of the calls.

Opposing this advantage, the new arrangement of keys introduces procedural changes that increase the length of many calls. For example, for the current workstation, the key pressed when a call is complete (POS-RLS) is on the far left of the keyboard and is typically pressed with the left index finger. Because most other keys are pressed with the right index finger, the TAO can move to the POS-RLS key with the left hand while other keys are being pressed with the right hand. The proposed workstation locates the POS-RLS key near the numeric keypad, with the other function keys, on the right side of the keyboard. Because of the POL-RLS key's proximity to the right hand, when we modeled this keystroke we assumed that, rather than making an awkward cross-body movement with the left hand, the TAO would press this key with the right index finger. This means that the horizontal movement to the POS-RLS key can no longer be done in the slack time while other keys are being pressed but must wait until the last function keypress is finished. This procedural change puts the movement to the POS-RLS key onto the critical path of several call types, increasing the overall length of those calls.

Screen Layout. Getting information from the screen involves moving the eyes to the correct location, waiting for information to appear if it is not already present, and perceiving and understanding the information. The need to wait for information to appear is a property of the system response time and is discussed later. The CPM-GOMS models assume that the TAOs are experts at finding the information they need from the screen; they can move their eyes to the next correct location in anticipation of getting the necessary information. Therefore, eye movements never appear on the critical path for either workstation. Likewise, the assumption of expertise means that the time to perceive and comprehend the information on both workstations can be estimated with either a complex visual-perception operator of approximately 290 msec or a binary visual-perception operator of approximately 100 msec (John, 1990). Although these operators are often on the critical path of both sets of models, because they are the same for both workstations, they do not produce a difference in the call-length predictions.

Keying Procedures. For several calls, the keying procedures for the proposed workstation eliminated keystrokes. In some of these calls, this decrease in keystrokes was an advantage for the proposed workstation. However, because of the complex interaction of parallel activities in the TAO's task, merely eliminating keystrokes is not necessarily an advantage. For example, Figures 19 and 20 show the first and last segments of a CPM-GOMS analysis for a calling-card call in which new procedures eliminated two keystrokes from the beginning of the call and added one

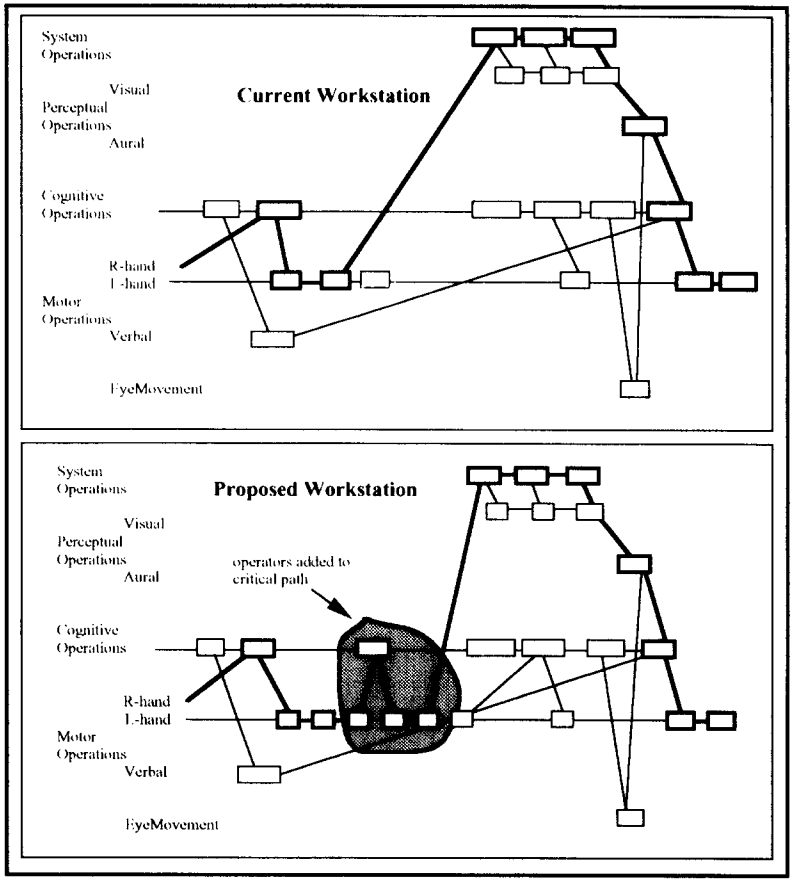
Figure 19. **Section of CPM-GOMS analysis from near the beginning of the call. Notice that the proposed workstation (bottom) has removed two keystrokes (which required seven motor and three cognitive operators) from this part of the call. However, none of the 10 operators removed was along the critical path (shown in bold).**



keystroke to the end of the call, for a net decrease of one keystroke. For each figure, the top chart represents the call using the current workstation and the bottom shows the CPM-GOMS analysis for the same call using the proposed workstation.

Figure 19 has two striking features. First, the model for the proposed workstation has 10 fewer boxes than the model for the current workstation, representing two fewer keystrokes. Second, none of the deleted boxes is on the critical path; all are performed in slack time. At this point in the task, the critical path is determined by the TAO greeting and getting information from

Figure 20. **Section of CPM-GOMS analysis from the end of the call. Notice that the proposed workstation (bottom) has added one keystroke to this part of the call, which results in four operators (three motor and one cognitive) being added to the critical path (shown in bold).**



the customer. The CPM-GOMS model predicts that removing keystrokes from this part of the call will not affect the TAO's worktime. Worktime is controlled by the conversation, not by the keystrokes, and not by the ergonomics of the keyboard.

The middle of the model, not shown (the activities between those shown in Figures 19 and 20), is identical for both workstations and essentially shows the critical path being driven by how fast the customer says the 14-digit number to which the call should be billed. TAOs are taught to "key along" with the customer. Although a rapidly speaking customer could force the critical path to be determined by the TAO's keying speed, both workstations use the

standard numeric keypad, so the critical path (and resulting speed of keying in numbers) would be the same for both workstations.

If the proposed keying procedures simply eliminated the two keystrokes required by the current workstation in the beginning of the call, then CPM-GOMS would predict equivalent performance. For the proposed workstation, however, the procedure was changed so that one of the keystrokes eliminated at the beginning of the call would occur later in the call (see the four extra boxes in the bottom of Figure 20). In this model, this keystroke goes from being performed during slack time to being performed on the critical path. The cognitive and motor time required for this keystroke now adds to the time required to process this call. Thus, the net elimination of one keystroke actually increases call time because of the complex interaction between parallel activities shown in the critical-path analysis.

System Response Time. Time to display information to the screen as well as time to output keystrokes vary between workstations and generally contribute to the slowness of the proposed workstation. For example, the proposed workstation is slower than the current workstation in displaying the first line of information but faster to display an entire screenful of information. In some call types, the information displayed at the bottom of the screen is on the critical path, and this speed-up in display time provides an advantage for the proposed workstation. However, the very first line of information is necessary for the TAO to decide which greeting to use to initiate the call, and waiting for this information to be displayed is on the critical path of every call. The manufacturer's estimate of the latency to display that first line of information is 0.57 sec longer for the proposed than for the current workstation. This half second is added to the duration of every benchmark call.

A less straightforward factor is the outpulsing of keystrokes. The proposed workstation is faster in outpulsing large numbers of keystrokes (e.g., a 14-digit calling-card number) but slower to outpulse single function keys. Whether this factor, number of keystrokes, favors, hurts, or is neutral to the worktime of the proposed compared to the current workstation depends only partly on how many keystrokes are being outpulsed. More important than number is what else is going on in the call while the numbers are being outpulsed. If the outpulsing is on the critical path, then number is an important factor; if it is not on the critical path, then number of outpulsed keystrokes does not matter.

Summary: Effect of Design Changes. The complex interactions among all these changes produced critical paths that were longer for the proposed than for the current workstation. The predominant reason was an increase in initial system response time, but time to outpulse single function keys, keying

procedures, and keyboard changes also contributed to the performance deficit.

Looking for a Learning Curve

> "Is there any point to which you would wish to draw my attention?"
> "To the curious incident of the dog in the night-time."
> "The dog did nothing in the night-time."
> "That was the curious incident," remarked Sherlock Holmes.
> (Doyle, 1892/1986, p. 472)

As discussed before (Section 3.2), worktimes on the proposed workstations reached asymptote during the first month of the trial. Performance during Months 2 through 4 was stable, with no evidence of improvement. This early asymptote surprised us. To try to explain this curious incident, we turn to the CPM-GOMS models with two questions. First, the worktime differences between workstations fluctuated during Months 2, 3, and 4. Did CPM-GOMS models predict this fluctuation? Second, because it is well known that skilled learning follows the power law of practice (Newell & Rosenbloom, 1981), do the CPM-GOMS models explain why the expected increased proficiency with the proposed workstation's keyboard and displays does not translate into the expected decrease in worktime?

Predicting the Fluctuation. During the trial, the monthly fluctuation in the proposed versus the current workstation difference (7.1%, 2.4%, 5.2%, and 2.5%) led some to argue that if the trial had continued long enough, eventually the proposed TAOs would be faster than the current ones. Although we believe that the explanations given before effectively show why the proposed has to be slower, we are interested here in whether the CPM-GOMS models could have predicted some of this monthly fluctuation.

For each month, Figure 21 shows the percentage by which the proposed workstation was slower than the current workstation. To obtain the trial data, for each month we collapsed over TAOs to find the median worktime for each call category and weighted these by the number of occurrences of that call category.

For the CPM-GOMS plot, we weighted the CPM-GOMS worktime prediction for each call category by the number of occurrences of that call category for that month. Because the CPM-GOMS models assume peak expert performance with no by-month variations in their worktime predictions, our by-category CPM-GOMS worktime estimates were constant for each month.

The actual monthly difference in workstation time, labeled "Trial" in

Figure 21. **Predicted versus actual monthly fluctuations in worktime differences between current and proposed workstations.**

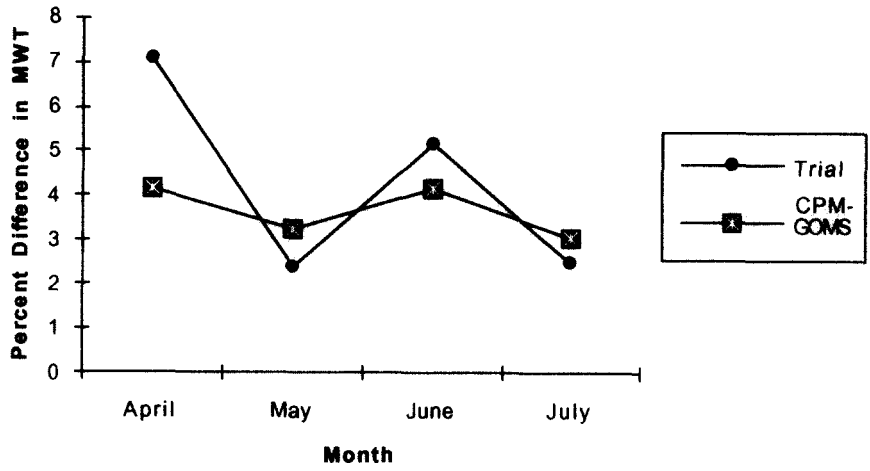


Figure 21, varies: 7.1%, 2.4%, 5.2%, and 2.5%. These fluctuations in the empirical data are suggested by the CPM-GOMS predictions (4.2%, 3.2%, 4.2%, and 3.1%).

Other than the first month, the CPM-GOMS models predict a fluctuation in monthly performance with the same overall pattern as the by-month empirical analysis. This result demonstrates that, because the workstation difference is not constant but varies with call category, monthly performance fluctuations can arise from changes in the call-category mix. This is consistent with our conjecture that the proposed group reached asymptotic performance within the first month.

Would Learning Have Been Noticed? An examination of the critical path of the proposed models suggests that we would not have noticed learning even if it had occurred. Before the trial, all of our TAOs were expert in handling calls. For the proposed workstation, all they had to do was learn the new keyboard and display layouts and some differences in keying procedures.

Assume that the cognitive operations that recall the procedures, and the primarily motor operations that search for newly positioned screen information and function keys, take substantially longer than their counterparts at asymptote. A workstation novice (but expert call handler) would show a different critical path and longer worktimes than the CPM-GOMS models used here. However, these operations would speed up with practice, and many of them would eventually crossover from being on the critical path to being in the slack time. For example, the first eye movement in the expert model might be a longer visual search for a workstation novice. However, that

visual search would cross over slack time as soon as it dropped below 750 msec. Likewise, as soon as the search for the new function key to release the call in the example credit-card call dropped below 2 sec, this keystroke would disappear into the slack time. Thus, our data and the power law of practice can be reconciled. The TAOs probably did get faster at using the new keyboard and reading from the new displays, but after an initial period, when the new operations were on the critical path, any additional learning would disappear into the slack time. CPM-GOMS alone cannot predict when that crossover would happen, but it can predict that, when it does, further practice effects would not be observed.

5. IMPLICATIONS FOR DESIGN

The CPM-GOMS models for the proposed workstation were specification based; that is, we built the models as if the proposed workstation did not exist other than as a list of engineering and ergonomic specifications. The previously presented data and models allow us to conclude that the CPM-GOMS models very accurately predicted the results of a large-scale empirical trial and that the specification-based models for the proposed workstation were as accurate as the observation-based models of the current workstations. We believe this result has implications for the design of future systems. We concur with Newell and Card (1985, p. 214) that "design is where the action is" and that the biggest gain from CPM-GOMS models will come from using them not just to evaluate completed designs but as tools in the design process. We envision an iterative process in which the designer first uses models qualitatively to focus design effort by suggesting areas in which performance could be improved. As ideas emerge, models would be used quantitatively to predict worktime differences among alternative designs. The designer could then confer with various other specialists to determine if the potential worktime savings justify the cost of implementing the design alternatives. Several examples follow.

5.1. Focusing the Design Effort

Every computer programmer lives by the maxim "profile before optimizing." That is, before expending the programming effort to optimize a section of code, determine how much runtime is associated with that code. With this information, trade-offs between expected effort and expected benefits can be made. The CPM-GOMS models provide a way to profile the component activities in a complex interaction between humans and machines. For example, Figure 22 shows such a profile for the percent of worktime consumed by waiting for the system response time, talking to the customer

Figure 22. **Percent time that various activities are on the critical path for the current workstation.**

Call Category	System Response Time	Talking	Keying	Reading	Ring/Coins
cc01	25%	40%	1%	3%	31%
cc02	3%	93%	0%	4%	0%
cc03	20%	71%	2%	6%	0%
cc04	25%	31%	6%	4%	35%
cc05	19%	44%	3%	2%	22%
cc06	12%	79%	2%	6%	0%
cc07	30%	57%	8%	3%	0%
cc08	26%	41%	23%	10%	0%
cc09	13%	65%	15%	7%	0%
cc10	9%	83%	2%	6%	0%
cc11	26%	63%	4%	3%	3%
cc12	11%	75%	6%	8%	0%
cc13	7%	89%	0%	4%	0%
cc16	15%	55%	3%	2%	25%
cc18	1%	75%	12%	2%	0%
Average	16%	64%	6%	5%	8%

(Talking), keying in numbers or function keys (Keying), reading information from the screen (Reading), and waiting while the called number rings or the customer deposits coins (Ring/Coins). The numbers here are not the actual time spent in each one of these activities but rather the percentage of time these activities are on the critical path for the benchmarks on the current workstation.

One implication of Figure 22 is that, with the current workstation as a baseline, there is little potential worktime savings from redesigning either the keyboard or the display. As a boundary condition, if the duration of keystrokes was reduced to zero, the most savings that could be expected would be 6% or, likewise, 5% for reading information from the screen. (Actually, the CPM-GOMS models would predict even less savings than these, because reducing the durations of the keying or reading operators to zero would change the critical path so that other activities would add time to the length of the call.) More substantial savings could come from reductions in system response times (up to an average of 16% at the boundary).

The most striking feature of Figure 22 is that conversation with the customer is the dominant activity on the critical path in almost every benchmark call (except cc04, in which ringing dominates slightly). It is illuminating to reflect on the data in Figure 22 in view of the way that laboratory evaluations and field experiments are typically conducted. In

typical laboratory studies, the focus is on designing the computer-human interaction, and the factors of greatest interest are the physical design of keyboards (and similar components) as well as the format and content of the displays. As our data show, focus on these factors will have extremely small impact on the system in use, although the impact in the laboratory may be statistically significant. In typical laboratory experiments, the focus is on proper experimental control. Because, in the case of TAOs, conversations with customers can neither be measured as a dependent variable nor controlled as an independent variable, this aspect is factored out of the experimental design. As a result, such experiments overlook the most important aspects and have little real-world validity. We conjecture that the proposed workstation was designed and evaluated in these traditional ways, and, although the workstation did well by traditional measures, the process missed the significance of the human–human interaction. Further, we conjecture that the use of traditional design and evaluation methods, which overlook practically significant factors, is far too common in the design of computer-based systems.

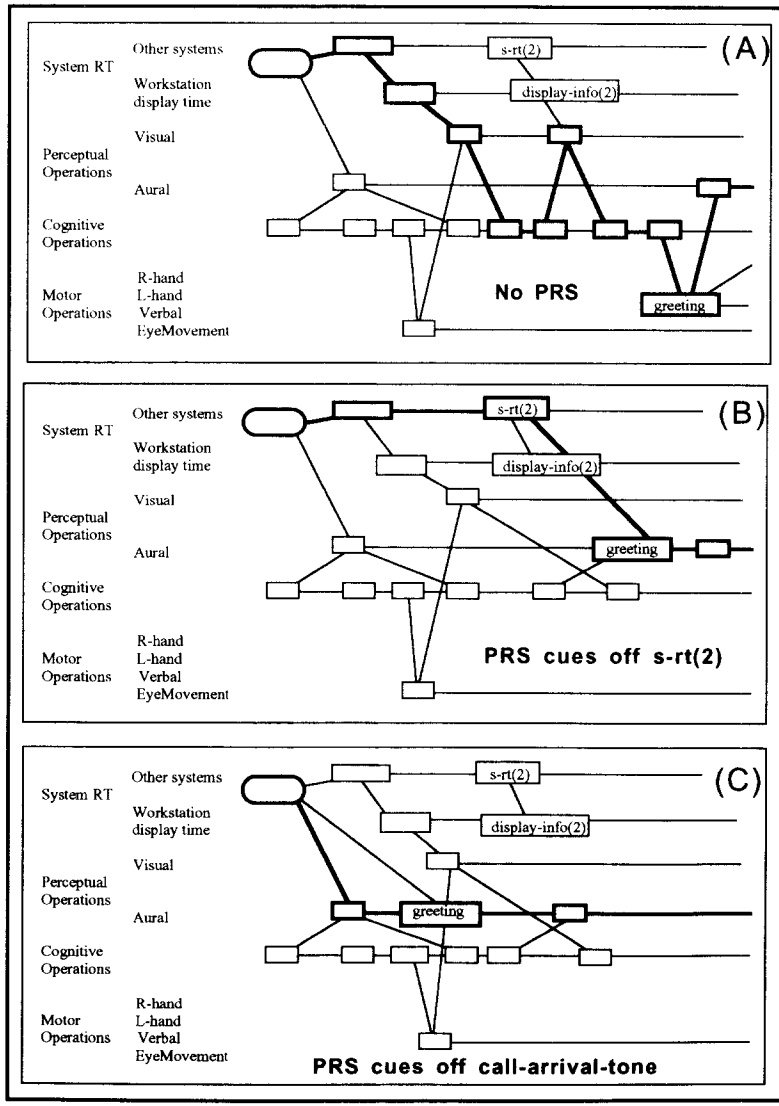
An interesting design philosophy would be to focus not on the hardware or software of the telephone system but on the conversation (Lawrence & Dews, 1992). When the task is viewed from this angle, the telephone company might introduce customer-education programs to inform customers how to provide information that will shorten call handling. For instance, on a collect call, during the slack time incurred while waiting for the called party to answer, the TAO could say a phrase such as, "In the future, if you provide your name first when making a collect call, it will speed your call" (the phrase takes approximately 3 sec to say whereas even two rings of the called phone take more than 5 sec to complete).

5.2. Quantitative Evaluation of Design Ideas

As new systems are proposed, routine tasks performed on these systems can be modeled. As in this study, modeling could be limited to a small number of benchmark tasks selected to represent high-volume activities, possibly low-frequency emergency procedures or low-frequency/high-profit (or high cost) tasks.

As an example of quantitative evaluation, consider using CPM-GOMS to evaluate the possible time savings from adding a personal response system (PRS) to the TAO's workstation. A PRS is a recording of the TAO's voice that automatically greets the customer with the proper phrase, for example, "New England Telephone, may I help you?" Currently (see Appendix A and Figure 23A), the TAO cannot initiate the greeting until INFO(2) is displayed, perceived, and verified.

Figure 23. **Portion of the CPM-GOMS that includes the TAO's greeting to the customer, "New England Telephone, may I help you?" (A) The current situation in which the TAO indicates the greeting after perceiving and verifying INFO(2). (B) The CPM-GOMS chart for a personal response system (PRS) that cues off of the system-rt that displays INFO(2). (C) A PRS that cues off of the call-arrival tone.**



Assume that the PRS greeting would take the same amount of time as the TAO's own voice. If the PRS started when the workstation displayed the same information that the TAO presently uses to choose a greeting (Figure 23B), then the CPM-GOMS models would predict a time savings of one complex visual-perception operator and two cognitive operators for approximately 390 msec. This translates into a potential cost savings of $1.2 million a year. If the PRS cued off of something earlier in the call processing, such as the information that cues the current call-arrival tone (Figure 23C), then an additional 500 to 1,000 msec could be saved (depending on the load of calls on the network) for an additional potential savings to NYNEX of $1.5 to $3 million per year. The estimated potential savings could be used to decide whether to invest in the development of a PRS.

5.3. Sensitivity Analysis of System Response Time

Sensitivity analysis can be used to help designers weigh the sensitivity of total time to changes in the duration of one operator or of one class of operators (Card et al., 1983). For example, at the very beginning of the call, the system response time, **system-rt(1)** (Figure 7 and Appendix A), of the proposed workstation is 570 msecs slower than that of the current workstation. This first system response time displays information that the TAO requires to initiate the proper greeting[10] and is on the critical path.

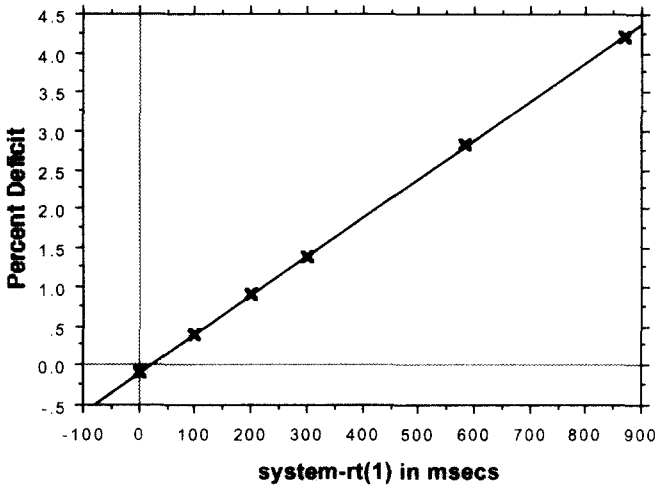
The engineering costs of reducing **system-rt(1)** must be weighed against the expected benefits. As an example, we have taken the CPM-GOMS model for cc07 of the proposed workstation and plotted predicted changes in proposed deficit against changes in duration of **system-rt(1)** (see Figure 24).

The predicted **system-rt(1)** for this call is 870 msec. This estimate is 570 msec above the 300 msec observed for the current workstation in the videotaped benchmark for this call and is based on estimates supplied by the manufacturer (see Figure 8). The total predicted deficit for the proposed versus the current workstation on this call is 4.3% (this deficit is a combination of the slower initial **system-rt(1)** as well as factors that occur later in the call).

As **system-rt(1)** is reduced, the deficit is reduced (Figure 24). At 300 msec (the **system-rt(1)** for the current workstation), a 1.4% deficit still exists because of other changes to the workstation (keyboard, procedures, other system response time changes, etc.). If **system-rt(1)** could be reduced to 0 (i.e., eliminated completely), the proposed deficit would transform into a 0.1% advantage.

[10] We assume for this example that the time to display this first piece of information is independent of any other system response time so that any engineering change that increases or decreases **system-rt(1)** would be neutral with respect to all other system response times.

Figure 24. **For cc07, the percent deficit, [(Proposed – Current)/Current] × 100, of the proposed workstation for different durations of system-rt(1): 0, 100, 200, 300, 585, and 870 msec.**



Engineers looking at such a chart would have to weigh the relative costs of each incremental decrease in system-rt(1) against the obtained benefits. Alternatively, designers and engineers might decide that the greatest cost effectiveness would come from leaving system-rt(1) alone and concentrating their efforts on other changes (e.g., installing a PRS that cues off of the begin-call information; see Figure 23).

6. CONCLUSIONS

This study validates the use of CPM-GOMS in predicting performance time for routine cognitive tasks accomplished through complex interaction among a user, a workstation and associated computer systems, and another human. In addition to quantitatively predicting the outcome of the field trial, the CPM-GOMS models provided explanations for the results. Indeed, the CPM-GOMS models saved the field trial from a potential disaster. Faster display times and the elimination of keystrokes from most calls were expected to result in faster worktimes. The data from the trial were so counterintuitive that, in the absence of a compelling explanation as to why the proposed workstation was slower than the current one, the tendency was to blame the trial instead of the workstation (Atwood et al., in press). On the basis of these analyses, NYNEX decided not to buy the proposed workstation.

The accurate prediction of workstation performance challenges the efficacy

of conducting large, empirical field trials. In many cases, the time and expense required for empirical trials might be eliminated and replaced by the much shorter and less disruptive time required for model building. In other cases, the models can be used to sort among competing devices and procedures. For example, if there were five workstations to be evaluated, CPM-GOMS might predict that two of these were better than the others, with small differences between the two best. An empirical trial could then be conducted to evaluate factors such as hardware quality and maintenance. In this way, CPM-GOMS would allow us to thoroughly consider more workstations in less time than is currently possible.

The explanations provided by the CPM-GOMS models far surpass any information given by empirical trials alone. These explanations led to an understanding of why old technology can outperform new technology, why this difference was not constant but varied with call category, and why learning over the course of the trial did not affect worktime. The CPM-GOMS models allow us to see the forest for the trees, evaluating all components of the task as an integrated whole with complex interactions. This holistic approach, dictated by the CPM-GOMS analysis methodology, is of value in understanding the complex effects of any design decision.

We believe that the biggest benefit will be to use the explanatory and predictive powers of CPM-GOMS in the design process to focus design effort and to provide a quantitative test-bed for design ideas. Such an approach is currently being used at NYNEX (Stuart & Gabrys, 1993), and it represents the early fruits of the decade-long struggle to "harden the science" of human-computer interaction (Newell & Card, 1985, p. 237).

Acknowledgments. The first two authors contributed equally to this article. The order of their names reflects alphabetical order and not seniority of authorship. We thank Karen O'Brien of the NYNEX Telesector Resources Group for her help in understanding the TAO's task and for her unfailing support in both the theoretical and the empirical portions of this project; Sandy Esch for her tireless efforts in transcribing the benchmark videotapes and for her vigilance in keeping our models conforming to what was actually observed in the tapes; Deborah Lawrence and Rory Stuart for their role in trial design and data collection; Karen O'Brien and Althea Turner for assisting in the review and revision of the CPM-Models for the current workstation; Allen Newell for his encouragement and insights into how GOMS analyses apply in the telephone-operator domain; Roy Taylor, Rory Stuart, Karen O'Brien, and Jean McKendree for their comments on earlier drafts; John Houtz for his advice and assistance in helping us think through some of the issues involved in dealing with outliers and statistically suspicious call categories; and Peter Polson and Stuart Card for their extremely detailed reviews.

Support. Bonnie John's participation was supported, in part, by the Office of Naval Research, Cognitive Science Program (Contract No. N00014-89-J-1975N158). The views and conclusions contained in this document are those of the

authors and should not be interpreted as representing the official policies, either expressed or implied, of the Office of Naval Research or of the U.S. Government.

REFERENCES

Atwood, M. E., Gray, W. D., & John, B. E. (in press). *Project Ernestine: Analytic and empirical methods applied to a real-world CHI problem.* San Mateo, CA: Morgan-Kaufmann.

Benigni, D., Yao, S., & Hevner, A. R. (Eds.). (1984). *A guide to performance evaluations of database systems* (National Bureau of Standards Special Publication 500-118). Washington, DC: U.S. Government Printing Office.

Benigni, D., Yao, S., & Hevner, A. R. (Eds.). (1985). *Benchmark analysis of database architectures: A case study* (National Bureau of Standards Special Publication 500-132). Washington, DC: U.S. Government Printing Office.

Bovair, S., Kieras, D. E., & Polson, P. G. (1990). The acquisition and performance of text-editing skill: A cognitive complexity analysis. *Human-Computer Interaction, 5,* 1-48.

Card, S. K., Moran, T. P., & Newell, A. (1980a). Computer text editing: An information processing analysis of a routine cognitive skill. *Cognitive Psychology, 12,* 32-74.

Card, S. K., Moran, T. P., & Newell, A. (1980b). The keystroke-level model for user performance time with interactive systems. *Communications of the ACM, 23,* 396-410.

Card, S. K., Moran, T. P., & Newell, A. (1983). *The psychology of human-computer interaction.* Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Doyle, A. C. (1986). Silver blaze. *Sherlock Holmes: The complete novels and stories* (Vol. 1, pp. 455-477). New York: Bantam. (Original work published 1892)

Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology, 47,* 381-391.

Gray, W. D., John, B. E., & Atwood, M. E. (1992). The précis of Project Ernestine or an overview of a validation of GOMS. *Proceedings of the CHI '92 Conference on Human Factors in Computing Systems,* 307-312. New York: Addison-Wesley.

Gray, W. D., John, B. E., Lawrence, D., Stuart, R., & Atwood, M. E. (1989, May). *GOMS meets the phone company, or, can 8,400,000 unit-tasks be wrong?* Poster presented at the CHI '89 Conference on Human Factors in Computing Systems, Austin, TX.

Gray, W. D., John, B. E., Stuart, R., Lawrence, D., & Atwood, M. E. (1990). GOMS meets the phone company: Analytic modeling applied to real-world problems. *Proceedings of Human-Computer Interaction—INTERACT '90,* 29-34. New York: Elsevier.

Hillelsohn, M. J. (1984). Benchmarking authoring systems. *Journal of Computer-Based Instruction, 11,* 95-97.

John, B. E. (1988). *Contributions to engineering models of human-computer interaction.* Unpublished doctoral dissertation, Carnegie Mellon University, Pittsburgh.

John, B. E. (1990). Extensions of GOMS analyses to expert performance requiring perception of dynamic visual and auditory information. *Proceedings of the CHI '90 Conference on Human Factors in Computing Systems,* 107-115. New York: ACM.

John, B. E., & Gray, W. D. (1992, May). *GOMS analyses for parallel activities.* Tutorial presented at the CHI '92 Conference on Human Factors in Computing Systems, Monterey, CA.

John, B. E., & Newell, A. (1989a). Cumulating the science of HCI: From S-R compatibility to transcription typing. *Proceedings of the CHI '89 Conference on Human Factors in Computing Systems,* 109-114. New York: ACM.

John, B. E., & Newell, A. (1989b). Toward an engineering model of stimulus-response compatibility. In R. W. Proctor & T. G. Reeve (Eds.), *Stimulus-response compatibility: An integrated perspective* (pp. 427-479). New York: Elsevier.

John, B. E., Rosenbloom, P. S., & Newell, A. (1985). A theory of stimulus-response compatibility applied to human-computer interaction. *Proceedings of the CHI '85 Conference on Human Factors in Computing Systems,* 212-219. New York: ACM.

Judd, C. M., & McClelland, G. H. (1989). *Data analysis: A model-comparison approach.* New York: Harcourt Brace.

Keppel, G. (1973). *Design and analysis: A researcher's handbook.* Englewood Cliffs, NJ: Prentice-Hall.

Lawrence, D., & Dews, S. (1992, May). *Natural dialog in a time-sensitive setting: A study of telephone operators.* Short talk presented at the CHI '92 Conference on Human Factors in Computing Systems, Monterey, CA.

Lerch, F. J., Mantei, M. M., & Olson, J. R. (1989). Skilled financial planning: The cost of translating ideas into action. *Proceedings of the CHI '89 Conference on Human Factors in Computing Systems,* 121-126. New York: ACM.

Newell, A., & Card, S. K. (1985). The prospects for psychological science in human-computer interaction. *Human-Computer Interaction, 1,* 209-242.

Newell, A., & Rosenbloom, P. S. (1981). Mechanisms of skill acquisition and the law of practice. In J. R. Anderson (Ed.), *Cognitive skills and their acquisition* (pp. 1-55). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Norwine, A. C., & Murphy, O. J. (1938). Characteristic time intervals in telephone conversation. *Bell System Technical Journal, 17,* 281-291.

Olson, J. R., & Olson, G. M. (1990). The growth of cognitive modeling in human-computer interaction since GOMS. *Human-Computer Interaction, 5,* 221-265.

Roberts, T. L., & Moran, T. P. (1983). The evaluation of text editors: Methodology and empirical results. *Communications of the ACM, 26,* 265-282.

Snedecor, G. W., & Cochran, W. G. (1967). *Statistical methods* (6th ed.). Ames: Iowa State University Press.

Stevens, J. (1986). *Applied multivariate statistics for the social sciences.* Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Stuart, R., & Gabrys, G. (1993, May). *A speech compression proposal for directory assistance: GOMS predictions.* Short talk presented at the INTERCHI '93 Conference on Human Factors in Computing Systems, Amsterdam.

HCI Editorial Record. First manuscript received September 6, 1991. Revision received May 5, 1993. Accepted by Stuart Card. Final manuscript received June 2, 1993. — *Editor*

APPENDIX A. SAMPLE CPM-GOMS ANALYSIS

A sample CPM-GOMS schedule chart for a calling-card call on the current workstation is presented here. Each Model Human Processor-level operator is represented as a box with a name centered inside it and an associated duration (see Figure 8) above the top-right corner (in msec). Lines connecting the boxes represent information-flow dependencies; that is, when a line joins two operators, the operator to the left produces information required by the operator to the right. For visual clarity, we place operators of the same category along a horizontal line (see Section 2.1, The CPM-GOMS Analysis).

The critical path is the sequence of operators that, because of their durations and dependency relationship to other operators, determines the total time of the task (see Section 2.1, The Critical Path). In the sample schedule chart, the critical path is indicated by the bold outline of the operators' boxes and bold dependency lines between them. The sum of the durations of the operators on the critical path is the total time for the task (see Figure A-1).

Figure A-1A. **Sample CPM-GOMS analysis, from beginning of call.**

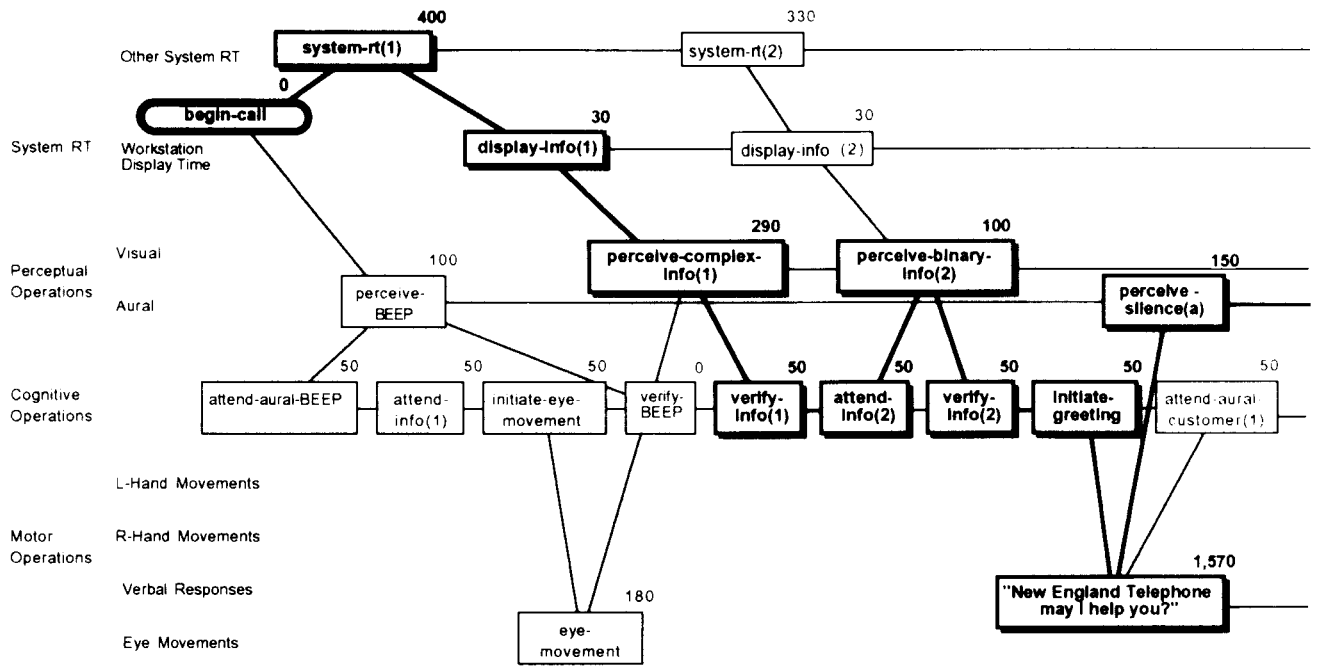


Figure A-1B. **Sample CPM-GOMS analysis (continued). Obtaining information from customer and initial keystrokes.**

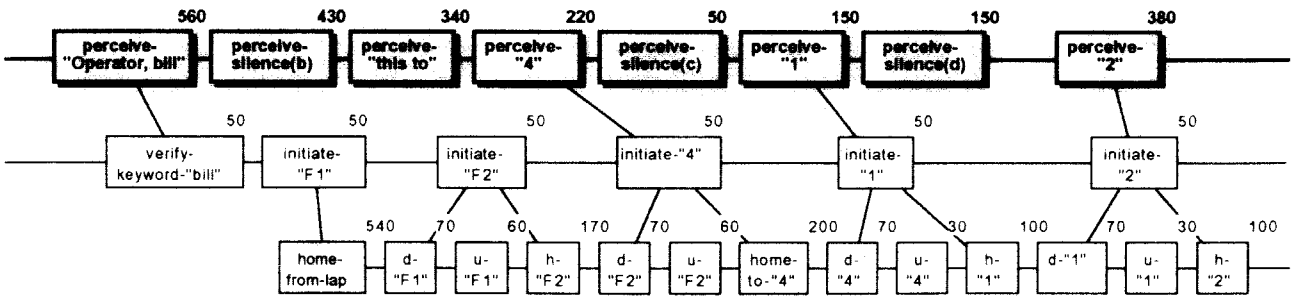


Figure A-1C. **Sample CPM-GOMS analysis (continued). "Keying along with the customer." Note that the critical path is determined by how fast the customer speaks, not by the TAO's keying rate.**

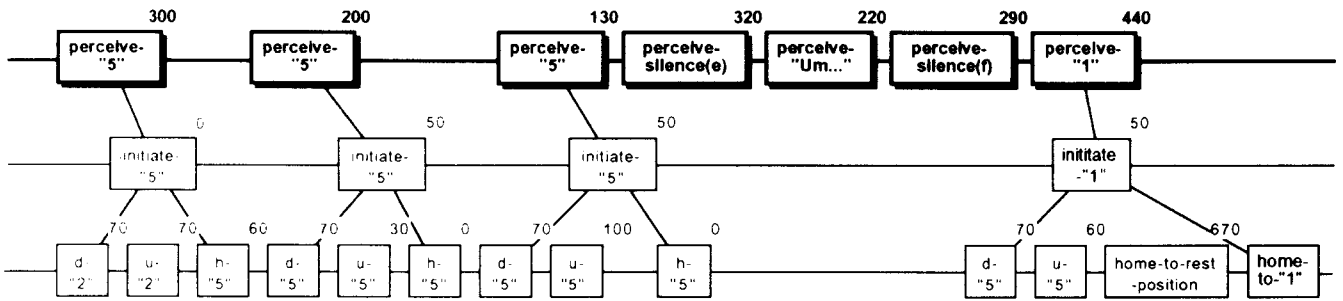


Figure A-1D. **Sample CPM-GOMS analysis (continued). "Keying along with the customer."**

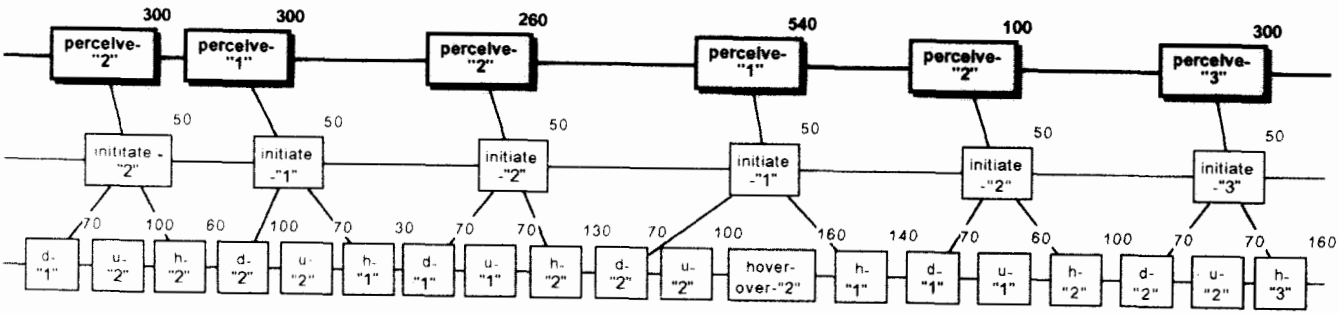
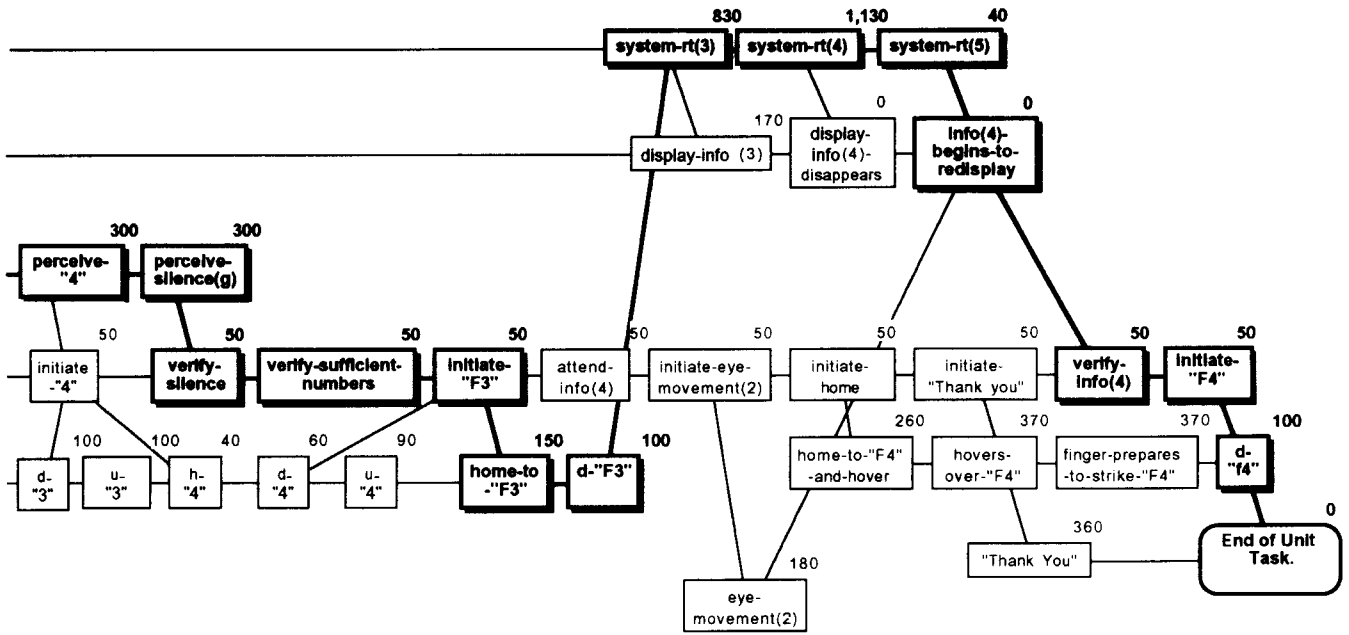


Figure A-1E. Sample CPM-GOMS analysis (continued), from end of call.



APPENDIX B. MISSING SUBJECTS, MISSING DATA, DROPPED CALL CATEGORIES, AND VARIABILITY BETWEEN CALL CATEGORIES

The 78,240 calls collected by the database software were unevenly distributed over the 3,840 cells of the field trial: 2 Groups × 24 Toll and Assistance Operators (TAOs) × 20 Call Categories × 4 Months. The unevenness of the distribution raised the concern that the worktime estimates derived from the empirical data for a given group for a given month for a given call category might not be a reliable estimate of the true worktime for that cell. We must stress that our concern here is not with the use of the empirical data in the statistical analyses but rather with the use of the empirical data to test the predictions of the CPM-GOMS models. Indeed, the major statistical test we use for analyzing the empirical data, the analysis of variance (ANOVA), is very robust and insensitive to just those violations of its assumptions that might occur from an uneven distribution of calls to conditions. Unlike comparisons among groups using empirical data, however, there are no accepted standards for judging the adequacy of a model's prediction of task-completion time as compared to empirical estimates of task time. With the exception of measures taken to compensate for missing subjects, the steps discussed shortly were taken with the aim of ensuring that the worktimes obtained from the empirical data were reliable and valid estimates against which to compare predictions of the CPM-GOMS models. After discussing these steps, we present a measure of relative variability, the coefficient of variation (CV), by which the reader may compare the differences in variability by group and by call category.

B1. Use of Median Worktimes

The use of the median score is a standard practice in reaction-time studies. Such studies are characterized by a positively skewed distribution that results from inherent limits on how fast a response can be made but no limits on how slow. The use of the median score is an important factor in reducing the variability when Time is the dependent factor, and it was used in this study. For example, for the current group, TAO 110, in April, for cc01 (where *cc* = call category), the database recorded 69 calls for which the median call was 24 sec. The median of the 69 calls is taken as the value of this cell. The median worktimes were computed for each of the 3,840 cells that contained data. These times were used to calculate means and standard deviations for all the analyses of the empirical data.

B2. Missing Subjects

During the last month of the trial, one of the TAOs was transferred to another job. To keep the design balanced, we dropped him and his matched control TAO from the analysis. Hence, all analyses reported are based on 23 subjects per condition.

B3. Missing Data and Call Categories

For purposes of this article and the validation of GOMS, we were interested in obtaining reliable and valid empirical data with which to compare the predictions of the CPM-GOMS models. To this end, we adopted a conservative procedure to identify and eliminate outliers as well as multiple and converging criteria to identify and eliminate call categories for which the empirical data may not have provided reliable and valid worktime estimates.

Step 0: Prescreening of Call Categories

Whereas most of the call categories were selected because of their frequency of occurrence, several low-frequency call categories were included because of their special interest to New England Telephone. For two of these categories, cc15 and cc20, the number of calls recorded was very low (40 and 1, respectively, see Figure B-1). Because each call category represents 184 cells (2 Conditions × 23 TAOs × 4 Months), most of the cells for these call categories were empty. cc15 and cc20 were dropped from the analysis.

cc19 posed a different problem. This call category was not well scripted so that the call that was staged and recorded on videotape (see Section 2) did not adequately represent the call category. To further confound matters, shortly after the trial, parts of the handling of this call were automated, and the TAOs' procedures for this call were completely changed. Because we had no basis on which to model this call (see Section 2.3), CPM-GOMS models could not be constructed for this call category, and it was dropped from further consideration.

Step 1: Identify Outliers

As discussed before, each call category had 184 cells (2 Conditions × 23 TAOs × 4 Months). For each call category, we calculated the mean and standard deviation and used these to identify scores that were both 3 *SD* above or below the call category mean and separated from the other scores by 1 *SD*. Scores meeting this dual criteria were dropped.

For example, for cc06, $M = 23.79$, $SD = 7.40$, $-3\,SD = 1.60$, $+3\,SD = 45.98$. The five slowest scores for cc06 are 44, 44, 47, 51, and 61. The scores

Figure B-1. **Call category occurrence frequency during the field trial. Note that the categories not used in the analysis are indicated by italics (see text for an explanation).**

Call Category	Current	Proposed	Total
cc01	9,238	8,579	17,817
cc02	7,758	7,385	15,143
cc03	2,385	2,364	4,749
cc04	1,720	1,740	3,460
cc05	2,468	2,432	4,900
cc06	1,845	1,804	3,649
cc07	2,345	2,328	4,673
cc08	1,383	1,339	2,722
cc09	1,949	2,054	4,003
cc10	2,313	2,111	4,424
cc11	1,596	1,421	3,017
cc12	984	1,015	1,999
cc13	1,056	1,088	2,144
cc14	*572*	*534*	*1,106*
cc15	*23*	*17*	*40*
cc16	741	706	1,447
cc17	*354*	*387*	*741*
cc18	650	601	1,251
cc19	*478*	*476*	*954*
cc20	*1*	*0*	*1*
Totals	39,859	38,381	78,240

47, 51, and 61 are more than 3 *SD* above the mean. However, 47 is less than 1 *SD* from its closest neighbor, 44, and was therefore retained. Likewise, 51 is less than 1 *SD* from 47 and was retained. However, 61 is more than 1 *SD* away from 51 and was dropped.

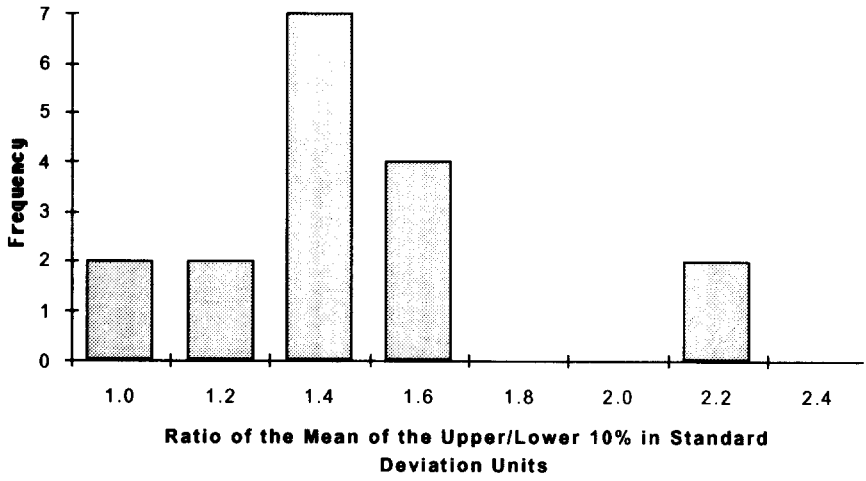
The criterion of ± 3 *SD* to identify an outlier was taken from Stevens (1986, p. 14). However, if the calls in some categories were nonnormally distributed (there is a limit to how fast a call can be successfully completed but apparently no limit to how slow), we did not want to use a procedure to identify outliers as a means to "normalize" a call category. This concern led to the adoption of the second criterion, that the call had to be separated from other scores by 1 *SD*.

This conservative procedure for identifying outliers resulted in 13 cells being dropped out of 3,128 cells representing 2 Conditions × 23 TAOs × 17 Call Categories × 4 Months.

Step 2: Identify Suspicious Call Categories

With the 13 scores identified in Step 1 dropped, the mean and standard deviation for each call category were recalculated, and our attention turned to

Figure B-2. **Distribution of standard deviation ratios for the 17 call categories.**



identifying and eliminating suspicious call categories. Following Judd and McClelland (1989), we avoided formal tests of normality, which these authors pronounced as having "serious disadvantages and problems" (p. 499). Rather, we used a procedure suggested by Houtz (personal communication, July 3, 1992) that had a simple and intuitive appeal and whose critical values emerged from applying the procedure to all the call categories and comparing the results. (We believe that this approach is within the spirit of that advocated by Judd & McClelland, 1989.)

For each category, the means for the upper 10% and lower 10% of the scores were determined. For example, for cc01, the 19 scores in the upper 10% had a mean of 25.42 whereas the 19 scores in the lower 10% had a mean of 20.92. These scores were then expressed as *z* scores, and their ratio was calculated. Continuing the example, for cc01, the mean (for all 184 scores) was 23.0 with a standard deviation of 1.34. The *z* score for the mean of the upper 10% was 1.80 ([25.42 – 23.00]/1.34), whereas the *z* score for the mean of the lower 10% was 1.55 ([23 – 20.91]/1.34). The ratio of upper to lower in standard deviation units was 1.16.

For data that are perfectly, normally distributed, the ratio of maximum to minimum in standard deviation units will be 1 (hence, the scores for cc01 are very close to the normal distribution). The distribution of this ratio from our 17 call categories is shown in Figure B-2.

As Figure B-2 shows, our adjusted (outliers removed, see Step 1) call categories show a fairly continuous range of ratios up until 1.6 *SD* units. After 1.6, there is a break, and then there are two call categories with ratios of around 2.2 *SD* units. These odd datapoints represent cc17 and cc14,

respectively. Additionally, an inspection of Figure B-1 shows that of the 17 call categories still under consideration these two have the fewest calls (741 for cc17 and 1,106 for cc14).

We view the break in the distribution of ratios as the "critical value" emerging from the data themselves. The fact that the two categories above this break also have the fewest calls adds to our suspicions that the empirical data may not be producing reliable and valid estimates of the worktimes for these categories. Accordingly, cc14 and cc17 were dropped from further analysis.

Step 3: Replacing Missing Data

With 15 call categories remaining, there are 2,760 cells in our analysis (15 Call Categories × 23 TAOs × 2 Groups × 4 Months). Of these cells, 12 were dropped in Step 1 (the 13th cell was in a call category, cc14, that was dropped in Step 2), and for nine cells there were no data. A total of 21 cells are missing (less than 1% of the 2,760).

For the Groups × Call Category × Months analysis, there are 120 (2 × 15 × 4) classes with 23 TAOs per class (120 × 23 = 2,760). The 21 missing observations are fairly evenly distributed among these 120 classes, with 2 classes missing data from two TAOs and 17 classes missing data from one. Each of the 21 missing observations was replaced by the mean of its class.

B4. Summary for Dropped Call Categories and Replacement of Missing Data or Outliers

The resulting 2,760 cells were used in the overall ANOVA reported in Section 3.2. Data based on these cells were used in Section 4.0 for comparison with the CPM-GOMS predictions.

Of the five call categories dropped, two were dropped for lack of data (cc15 and cc20), one for lack of an adequate benchmark script (cc19), and two for suspicion of nonnormality (cc14 and cc17). Fortunately, as Figure B-1 shows, these five call categories are the ones with the least data. The five call categories plus the 12 cells dropped reduce our number of calls from 78,240 to 72,390. Despite this reduction, more than 92% of the data are accounted for.

The important point is that although all 20 call categories may be of importance to NYNEX they are important here only to the extent that data collected for them, by group and by month, provide reliable and valid estimates of their median worktime under real-world conditions. The preceding procedure has left us with 15 call categories whose worktime estimates we trust. It remains to be seen whether the predictions derived from the CPM-GOMS models in Section 2 can be used in Section 4 to predict the variations in the empirical data that we describe in Section 3.

Figure B-3. **Coefficient of variation.**

Call Category	Current	Proposed
cc01	0.41	0.38
cc02	0.77	0.71
cc03	0.39	0.52
cc04	0.67	0.33
cc05	0.88	0.86
cc06	0.87	0.81
cc07	1.05	1.11
cc08	0.52	0.60
cc09	0.37	0.39
cc10	0.67	0.67
cc11	0.62	0.92
cc12	0.45	0.32
cc13	0.70	0.57
cc16	0.48	0.53
cc18	0.47	0.57

B5. Variability Between Call Categories

Figure B-3 lists the coefficient of variance (CV) for each call category for both groups. The CV is calculated by dividing a group's standard deviation[B-1] by its mean. The CV provides a measure of variation that is relatively uninfluenced by differences in the size of the means (Snedecor & Cochran, 1967). Additionally, because it "is the ratio of two averages having the same unit of measurement it is itself independent of the unit employed" (p. 64). We provide the CVs for those readers who wish to compare the variability among our categories or between our data and those reported by others (e.g., Card et al., 1983, pp. 159-160 and Figure 5.8).

[B-1] Note that estimates of standard deviation were derived from the same data we used for computing *z* scores (see footnote 8 for more details).