

Chapter 2

Performance Measurement and Evaluation in Human-in-the-Loop Simulations

Ling Rothrock

Abstract A prerequisite for designers of complex systems is a proper understanding of human performance characteristics. While human factors texts provide some insights into basic performance issues, the emergence of highly-automated computing systems have fundamentally altered the way humans work. The purpose of this paper is to present an approach to quantify and analyze human performance in human-in-the-loop simulations based on over ten years of research experience. The approach is centered on a measurement construct, called a time window, which enables a functional relationship between constraints on operator activities and time availability. A blackboard model is presented as the mechanism to generate, maintain, and complete time windows. To demonstrate the utility of time windows, an existing implementation in a real-time human-in-the-loop simulation is also described. An extension of time windows to measure team performance is also discussed. Using time window outcomes, samples of previous analyses are presented to exhibit the potential of the construct.

2.1 Foundations

2.1.1 Introduction

The emergence of highly-automated computing systems has fundamentally altered the way humans work. As these systems have increasingly become mediators between human operators and the work environment, human understanding of how

L. Rothrock (✉)

The Harold and Inge Marcus Department of Industrial and Manufacturing Engineering,
The Pennsylvania State University, Leonhard Building 210, State College,
University Park, PA 16802, USA
e-mail: lxr28@engr.psu.edu

work is accomplished has greatly diminished. Remarks of “What happened?” or “Why did it do that?” are not uncommon as operators seek to understand the processes of systems designed to improve their work. Rather than serving the purpose of being tools for human use, these systems have come to be regarded as autonomous agents to which humans must adapt in the workplace.

To investigate human decision making in these highly-automated systems, researchers have had to rethink the applicability of traditional laboratory methods such as expected utility theory (Beach and Lipshitz 1993). The use of traditional methods assumed that findings from the laboratory environment—where highly cognitive, single-choice tasks were conducted—could be applied to more realistic settings. The premise that findings from a static, forced-choice task can be extended to an operational environment has been called into question (Hammond 1986). In fact, some researchers have recommended that studies of human operators must occur in settings that are representative of the actual environment (Suchman 1987; Endsley 2006).

The purpose of this paper is to present a research approach to quantify and analyze human performance within a human-in-the-loop simulation based on over ten years of research experience. The key concept introduced here is the notion of a time window that provides a functional relationship between constraints on operator activities and time availability. A methodology is proposed to evaluate time windows as well as to assess operator attunement to them. This paper contains reprints of three journal articles. Section 2.1 is based on Rothrock (2001) that explains the foundations of time windows, Sect. 2.2 is based on Rothrock et al. (2009) that explores team performance measurement, and Sect. 2.3 is based on Ma et al. (2011) which extends performance measurement to service-based industries like call centers.

2.1.2 Situations, Constraints and Time Windows

2.1.2.1 Situativity Theory

In order to extract situations, constraints, and available time, these terms must first be clearly defined. The meaning of the terms “situation” and “constraint” as they have been used thus far is consistent with the interpretation provided by Greeno and Moore (1993) and Greeno (1998). They introduced a theory of situativity in which cognitive processes are analyzed as relations between operators and other subsystems in the environment. The theory is powerful because it stipulates that a functional relationship exists between an operator’s decision making activities and the task environment. The dependency relation between an action and the resultant situation—also known as a constraint—contains the following form (Greeno 1994, p. 339):

$$\langle\langle \text{action by operator} \rangle\rangle \Rightarrow \langle\langle \text{good effect in situation} \rangle\rangle,$$

where the good effects are outcomes that are required for a broader activity to be successful.

2.1.2.2 Time Window Extension to Situativity Theory

The notion of a time window is an extension to situativity theory. To computationally implement the time window extension, therefore, a greater degree of definitional precision is required. Accordingly, the definition of time windows conveys the concepts of situativity theory while relying upon temporal logic (Gabbay et al. 1994; Allen 1983) to provide the basic foundation for a computational model.

A time window is a construct that specifies a functional relationship between a required situation and a time interval that specifies availability for action. A time window does not specify what action must be taken, but only that there exists an action which will result in the required situation. In the course of operator activity within a dynamic task, n time windows are denoted as w_i for $i = 1-n$.

At the onset of operator interaction, all time windows are designated as inactive and represented by the set U_0 . Until a time window is designated as open, it remains inactive. Time windows are designated as open if the availability for action exists for a required situation at the current point in time space. The set of open time windows at time t is designated as O_t . When a required situation no longer exists, the corresponding time window is designated as closed. The set of closed time windows at time t is denoted as C_t . The membership of U , O , and C is defined to be persistent over time, and will remain the same (i.e., $U_{t+1} = U_t$, $O_{t+1} = O_t$, and $C_{t+1} = C_t$) unless designated otherwise. Methods to extract conditions specifying the opening and closing of time windows will be covered in Sect. 2.1.3.

To complete the constraint specified by situativity theory in a temporal context, one must define operator action and the relationship between action and time window. An operator action is defined here as a two-tuple that includes a detectable act performed by the operator at a specific point in time. In the course of operator interaction within a dynamic task environment, m actions are denoted as \mathbf{b}_j for $j = 1$ to m . The relationship between action and time window can be described by two Boolean indicator functions, I_w^l , such that, for $l = 1$, the function evaluates whether an action meets the required situation specified by a time window, and for $l = 2$, the function evaluates the relevance of an action toward a time window.

Thus,

$$I_w^1(\mathbf{b}) = \left\{ \begin{array}{ll} 1 & \text{if } \mathbf{b} \text{ meets situation specified in } w \\ 0 & \text{if } \mathbf{b} \text{ does not meet situation} \end{array} \right\}, \text{ and}$$

$$I_w^2(\mathbf{b}) = \left\{ \begin{array}{ll} 1 & \text{if } \mathbf{b} \text{ is relevant toward } w \\ 0 & \text{if } \mathbf{b} \text{ is not relevant toward } w \end{array} \right\}.$$

Six predicates, $M_T^k(w_i, \mathbf{b}_j)$ for $k = 1-6$, will now be constructed to characterize fundamental relationships between time windows and operators actions over a time interval T . In particular, the truth value, $||M^k(w_i, \mathbf{b}_j)||_{T+, T-}$, of each predicate

is evaluated for a time interval that starts when operator interaction in the task begins ($T+$) and ends when operator interaction ceases ($T-$). Given that \mathbf{b}_j occurs at time s , equations to evaluate the first five predicates are listed as follows:

- An on-time action that results in a required situation, $M_T^1(w_i, \mathbf{b}_j)$, is formally defined as,

$$||M^1(w_i, \mathbf{b}_j)||_{T+, T-} = 1 \text{ iff } \exists i \text{ such that } [I_{w_i}^1(\mathbf{b}_j) = 1] \wedge (w_i \in O_s); \quad (2.1)$$

- An early action that results in a required situation, $M_T^2(w_i, \mathbf{b}_j)$, is defined as,

$$||M^2(w_i, \mathbf{b}_j)||_{T+, T-} = 1 \text{ iff } \exists i \text{ such that } [I_{w_i}^1(\mathbf{b}_j) = 1] \wedge (w_i \in U_s); \quad (2.2)$$

- A late action that results in a required situation, $M_T^3(w_i, \mathbf{b}_j)$, is defined as,

$$||M^3(w_i, \mathbf{b}_j)||_{T+, T-} = 1 \text{ iff } \exists i \text{ such that } [I_{w_i}^1(\mathbf{b}_j) = 1] \wedge (w_i \in C_s); \quad (2.3)$$

- An action that is relevant toward a required situation, but does not result in it, $M_T^4(w_i, \mathbf{b}_j)$, is defined as,

$$||M^4(w_i, \mathbf{b}_j)||_{T+, T-} = 1 \text{ iff } \exists i \text{ such that } [I_{w_i}^1(\mathbf{b}_j) = 0] \wedge [I_{w_i}^2(\mathbf{b}_j) = 1]; \quad (2.4)$$

- An action with no corresponding time window, $M_T^5(\mathbf{b}_j)$, is defined as,

$$||M^5(\mathbf{b}_j)||_{T+, T-} = 1 \text{ iff } \forall i, (I_{w_i}^2(\mathbf{b}_j) = 0). \quad (2.5)$$

Because the sixth predicate is based on a time window instead of action, the equation to evaluate it is defined separately as follows:

- A time window that has been missed, $M_T^6(w_i)$, is defined as,

$$||M^6(w_i)||_{T+, T-} = 1 \text{ iff } \forall j, (I_{w_i}^2(\mathbf{b}_j) = 0). \quad (2.6)$$

Because of their reliance on temporal logic, Eqs. 2.1–2.5 offer a more explicit description of constraints than the conceptual distinctions offered by situativity theory. Specifically, the time window framework can now be utilized as a dependency relation between an action and a required situation that is also bound by time.

2.1.2.3 Extracting Time Window Information

To extract time window information, one must view operator decision making in its experiential context. The focus of the extraction is, therefore, on the use of analysis methods to discover mappings between operator actions and situations required to meet system objectives.

Three techniques meet the criteria for extracting time window information. Because each technique focuses on a slightly different information source, the most effective approach is one that integrates the advantages of all three. One method, cognitive task analysis (CTA) (e.g., Militello and Hutton 1998), is based on human input. CTA focuses on experienced practitioners in operational contexts to extract information they deem diagnostic to successfully operate in the task environment. The two other methods rely on theoretical and empirical studies of the environment in which the task is performed. Cognitive work analysis (CWA) utilizes theoretical expertise and engineering analyses of system dynamics to identify conceptual distinctions within a work domain that can later be used as modeling tools (Vicente and Rasmussen 1992). Ecological task analysis (ETA) is focused on analysis of the task environment to determine empirical regularities in environmental behavior (Kirlirk 1995). Time window information extracted through the integrated method should therefore be: valid from an operator's perspective; consistent with system dynamics; and true to the availability of action within the task environment. Consider, for example, the process of extracting time window information in an air traffic control (ATC) domain. CTA is used to determine normal operator courses of actions to reach established objectives. CWA is used to ascertain static and kinematic constraints in the ATC domain that affect the operator's ability to reach the objectives (e.g., radar range). ETA is used to discover constraints in the ATC environment (e.g., appropriate regulations) and empirical regularities to which good controllers must be sensitive.

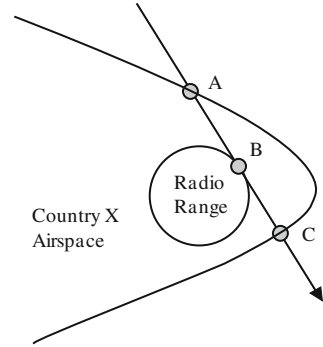
Once time window information has been extracted, the next step in the proposed research methodology is to implement the construct. The next section presents an object-oriented simulation architecture that includes a time window generation and maintenance system based on the blackboard model.

2.1.3 Blackboard Model in Object-Oriented Simulations

The blackboard model was first developed in the early 1970s as a tool for speech understanding (Erman et al. 1980). Since then, it has been implemented in many domains for multiple purposes. For example, Vranes et al. (1991) have used it as a tool to conduct military planning. Rubin et al. (1988) used it as a framework to construct an operator's associate in a supervisory control task. More recently, Adeli and Yu (1995) used it to develop an integrated computing environment to solve complex engineering problems. Although it has been implemented in vastly different forms, the blackboard model approach to problem solving remains the same. In essence, the blackboard model of problem solving is a reasoning scheme which applies pieces of knowledge at the most opportune time to construct a solution to the problem.

A blackboard model consists of three major components (Nii 1986): knowledge sources; the blackboard data structure; and control. The knowledge sources contain knowledge required to solve the problem. The blackboard data structure is a global

Fig. 2.1 Air traffic control example. An unknown aircraft enters Country X airspace at point A, enters and leaves range to establish radio contact at point B, and leaves Country X airspace at Point C



database in which partial and full solutions are kept. The blackboard control is an opportunistic reasoning model that guides problem solving by choosing and activating appropriate knowledge sources.

2.1.3.1 The Blackboard and Time Windows

To illustrate the use of blackboard model to open, maintain, and close time windows, consider the following example: In a real-time simulation, a human operator assumes the role of an ATC monitoring aircraft entering and leaving Country X's airspace (Fig. 2.1). The operator has been given specific instructions to identify all unknown aircraft entering the airspace, and to establish radio contact with all aircraft that come within radio range. An unknown aircraft, traveling along the trajectory indicated by the direction vector, enters Country X airspace at point A, enters and leaves range to establish radio contact at point B, and leaves Country X airspace at point C.

In the context of time windows, the blackboard knowledge sources include operators who act on the environment, and entities that produce situations. These sources contribute not only actions and situations to the blackboard, but also temporal information that defines constraints within the environment in which the task is performed.

In the example, the knowledge sources include the ATC and the unknown aircraft. Moreover, the unknown aircraft also reveals constraints that dictate expected ATC actions. At point A, w_1 is designated as open so that $w_1 \in O_{t_a}$ with the specification that the situation of a correctly identified aircraft be required. The time at which the aircraft reaches point A is designated as t_a . At point B, a second time window, w_2 , is designated as open to specify the situation of established radio contact at time t_b so that $w_2 \in O_{t_b}$. Since the trajectory of the aircraft is tangential to the curve bounding the radio contact area, the available time interval for the ATC to establish radio contact is instantaneous. Therefore, w_2 is also designated as closed at time t_b so that $w_2 \in C_{t_b}$. At point C, the aircraft exits Country X airspace and triggers the closing of w_1 so that $w_1 \in C_{t_c}$.

The blackboard data structure holds time window information in the form of computational and solution-state data. Each time window represents a structural means-ends hierarchy (Vicente and Rasmussen 1992) where the required situation (ends) is achieved by an expected operator action (means).

While the knowledge sources provide necessary information to generate and maintain time windows within the blackboard architecture, the activities on the blackboard are monitored and controlled by the blackboard control. The control uses opportunistic reasoning to apply backward reasoning as well as forward reasoning models to maintain time window information. Backward reasoning is applied at the point of a required situation to determine if the expected operator action has been taken, while forward reasoning starts at an operator action to determine if the action outcome meets any required situations.

Returning to the ATC example, assume that the operator takes three actions. The first action, \mathbf{b}_1 , incorrectly identifies the aircraft at time t_1 , where t_1 is before t_a (i.e., $t_1 < t_a$). The second action, \mathbf{b}_2 , correctly identifies the aircraft at time t_2 where $t_a < t_2 < t_c$. The third action, \mathbf{b}_3 , alerts Country X's border patrol at time t_3 where $t_b < t_3 < t_c$.

Using backward reasoning, the blackboard control examines all open time windows to determine if any has been met. At time t_a , the control assesses \mathbf{b}_1 as applicable to w_1 so that $I_{w_1}^2(\mathbf{b}_1) = 1$, but does not result in the required situation so that $I_{w_1}^1(\mathbf{b}_1) = 0$. Thus, Eq. 2.4 is satisfied and the action is deemed irrelevant. At time t_2 , the control determines that \mathbf{b}_2 is consistent with the expected operator action specified by w_1 so that $I_{w_1}^1(\mathbf{b}_2) = 1$. Moreover, because $w_1 \in O_{t_2}$, the control evaluates w_1 and \mathbf{b}_2 to satisfy Eq. 2.1 and assesses \mathbf{b}_2 an on-time, required action.

Applying forward reasoning, the control examines all current actions to determine if they address any required situations. At time t_3 , the control determines that \mathbf{b}_3 is not relevant toward any time window so that $\forall i, I_{w_i}^2(\mathbf{b}_3) = 0$. The control does not, however, make a determination on the action at this point. Rather, it seeks resolution of the action's status by checking backward reasoning results to ensure that the action is not early for a later time window. Nevertheless, the third action was eventually determined to be irrelevant.

2.1.3.2 Blackboard Models in a Real-Time, Object-Oriented Simulation

Conceptually, the use of time windows in a blackboard model has been demonstrated. To illustrate the utility of time windows in a simulation environment, the implementation of time windows via a blackboard model will now be presented. The simulation architecture developed at the Georgia Institute of Technology (Chu et al. 1991; Jones et al. 1995) is used as a baseline for discussion. The integration of the blackboard model within the simulation architecture is depicted in Fig. 2.2.

The active simulation object (ASO) is used as a base class so that events can be scheduled by methods contained in its subclasses. The display class contains

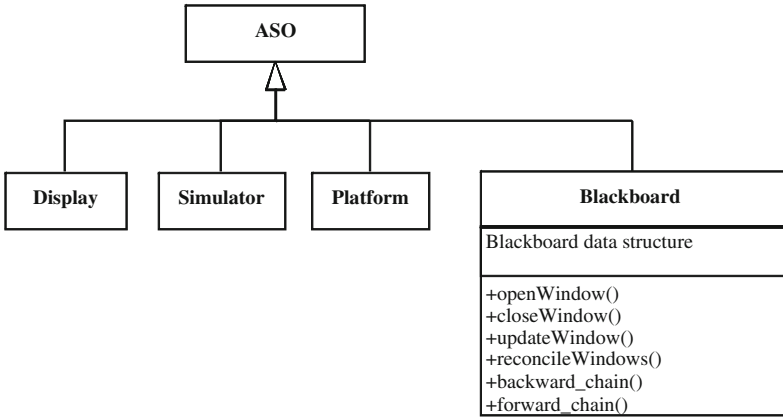


Fig. 2.2 Simulation class diagram

parameters as well as methods for generating the graphical user interface. The simulator class contains methods to control the experimental simulation. The platform class represents physical platforms (e.g., airplanes) that exist in the simulation environment, and contains methods that allow those objects to act upon the environment. The blackboard class contains the knowledge sources within the blackboard data structures. It also contains methods to control the blackboard by opening time windows, closing time windows, updating and reconciling time windows, conduct forward chaining reasoning, or execute backward chaining reasoning.

An illustration of the blackboard and time window implementation within an object-oriented simulation framework is represented in the form of a sequence diagram in Fig. 2.3. A sequence diagram is a model that describes how groups of objects collaborate in some behavior (Booch et al. 1999). Each box above the diagram represents an object. Each vertical line represents the object's life during the interaction. The flow of events is chronologically ordered from top to bottom. Methods labeled with an asterisk are iterative.

Revisiting the air traffic control example, the event flow of operator actions and aircraft movements is reflected in Fig. 2.3. A chronologically-ordered narration on the sequence of events follows:

1. The flight of the unknown aircraft along the southeasterly trajectory is accomplished by the iterative call of the `modifyPosition()` method.
2. The first operator action, \mathbf{b}_1 , of incorrectly identifying the aircraft (as a jet) is posted to the blackboard.
3. When the control detects the aircraft entering the airspace of Country X, w_1 is designated as open.
4. The backward-chaining model reasons that \mathbf{b}_1 is an incorrect identification that has been taken early. Thus, $\|M^4(w_1, \mathbf{b}_1)\|_{T+, T-} = 1$.

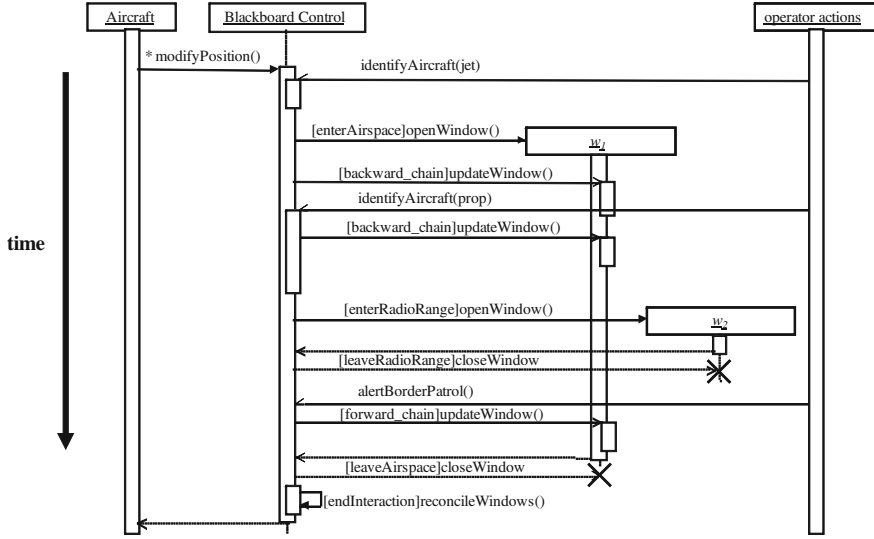


Fig. 2.3 Time window sequence diagram

5. The second operator action, \mathbf{b}_2 , of correctly identifying the aircraft (as a propeller-driven aircraft) is posted to the blackboard.
6. The backward-chaining model determines that a correct identification action has been taken on-time. Therefore, $\|M^1(w_1, \mathbf{b}_2)\|_{T_+, T_-} = 1$.
7. When the control detects the aircraft entering radio range, w_2 is designated as open.
8. The control immediately detects the aircraft leaving radio range, and closes w_2 .
9. The third operator action to alert the border patrol, \mathbf{b}_3 , is posted to the blackboard. The forward-chaining model determines that no time window specifies the need for \mathbf{b}_3 . Moreover, the action does not serve any required situation—radio contact or correctly identified aircraft. Therefore, the action is classified as irrelevant. Thus, $\|M^5(\mathbf{b}_3)\|_{T_+, T_-} = 1$.
10. When the control detects the aircraft leaving Country X airspace, w_1 is closed.
11. Operator interaction ceases as the aircraft leaves Country X airspace. At this point, the backward-chaining model reconciles the blackboard by closing all open windows and assessing if windows have been missed. The only window in question is w_2 , and is assessed to be missed so that $\|M^6(w_2)\|_{T_+, T_-} = 1$.

2.1.3.3 Possible Time Window Outcomes

The utility of a time window is not only in its temporal and functional descriptions, but also in the richness of the possible outcomes. Some time window outcomes have already been described. Not surprisingly, the complete space of possible time

		Environment				
Response	Situation Required			No Situation Required		
	Early	On-time	Late	Eq 5 False Alarm		
	Action	Correct	Eq 2		Eq 1	Eq 3
Incorrect		Eq 4				
No Action	Miss			Eq 6	Correct Rejection	

Fig. 2.4 Possible time window outcomes. The environment is delineated in terms of situation required (time window exists) or no situation is required (time window does not exist). Equations 2.1–2.4 represent actions that are relevant to a time window. Equations 2.1–2.3 represent actions that result in the required situation (correct actions). Equation 2.4 represents actions that do not meet the required situation (incorrect actions) even though they are relevant

window outcomes (Fig. 2.4) is represented by the fundamental relationships between time windows and operator actions outlined in Eqs. 2.1–2.6. In itself, the existence of a required situation does not impact system performance. It is the presence of operator action in a temporal context that specifies whether performance is good or poor. An incorrect, early action (first ATC operator action) is represented as Eq. 2.4. An on-time, accurate action (second ATC operator action) is represented as Eq. 2.1. An action with no corresponding required situation (third ATC operator action) is categorized as Eq. 2.5. A non-action for an existing situation requirement (no attempt to establish radio contact) is characterized as a miss and is represented as Eq. 2.6.

It has been shown that time window is a viable construct, both conceptually as well as in an implemented mechanism within a simulation framework. However, the value of implementing time windows in a research effort has yet to be discussed. The following section will discuss the implications of applying time windows toward human performance measurement and evaluation.

2.1.4 Time Windows and Human Performance

2.1.4.1 Implications Toward Measurement

Wickens and Holland (2000) observed that most performance measures are associated with one of the following categories of raw data:

- 1. Measure of speed or time (e.g., how fast can an operator reach for a lever?);
- 2. Measure of accuracy or error (e.g., how many typing mistakes are made?);

3. Measure of workload or capacity demands (e.g., how difficult is this task?); and
4. Measure of preference (e.g., is one display preferred over another?).

In most cases, the use of a particular type of measure is dependent on the real-world task to which the results of the laboratory task generalize. The emphasis, therefore, is on finding methods that analyze factors in isolation. However, it has already been noted that research on dynamic and complex environments should take place in representative settings. Recognizing the problem, researchers have sought to develop techniques to measure performance in tasks that are more representative of the operational environment. Sanderson et al. (1989) focused on the use of verbal protocol data in operational tasks. Howie and Vicente (1998) used automated log files to construct a number of measures to assess operator performance in a microworld setting. Still other researchers (Raby and Wickens 1994; Moray et al. 1991; Laudeman and Palmer 1995) focused on recorded data in time-critical task environment.

The time window construct represents a fundamental shift from existing performance measurement approaches. It is not focused solely on whether a certain task is completed, or how fast a certain button is pushed, or what percentage of error is detected. Rather, it provides a computational framework to dynamically evaluate heterogeneous situation demands and operator abilities to meet them in a complex domain. The benefit of the framework is the functional link between operator actions and the domain with which she/he interacts.

2.1.4.2 Implications Toward Evaluation

As shown in Fig. 2.5, utilization of the time window construct leads to a multi-dimensional space of possible outcomes. As yet, no mathematical formalism exists to comprehensively evaluate operator performance based on all dimensions. Instead, two methods are proposed to provide different perspectives on operator attunement to the constraints. The first method, factor analysis, is designed to determine correlations among different types of time windows and time window outcomes. The second method depends on the use of signal detection theory (SDT) to determine the sensitivity of operator actions to situation requirements.

Factor analysis is a data reduction technique that attempts to find a smaller number of dimensions, or factors, while retaining most of the information in the original space (Green 1978). The intent, therefore, is to evaluate which situations and operator actions can be aggregated into higher order factors. The analysis process proceeds in three major steps:

1. Rotate original data (i.e., variables consisting of the different time window outcomes in different types of required situations) to a new orientation that exhibits dimensions with maximal variance;
2. Reduce the dimension of the transformed data space; and
3. Identify the new dimensions, or factors, in terms of variables that show high association with each factor.

Fig. 2.5 Signal detection theory outcomes

	State of the world	
	Signal	Noise
Response		
Detected	Hit	False Alarm
Not Detected	Miss	Correct Rejection

The reader is referred to any multivariate statistics text for details on steps 1 and 2. To identify underlying factors, a technique called the scree test (Cattell 1966) is suggested. In essence, the scree test requires plotting the variance accounted by each factor extracted, and then finding elbow in the curve of the plot. To identify which variables belong to the selected factors, factor loadings (i.e., correlation between the variable with a factor) are recommended.

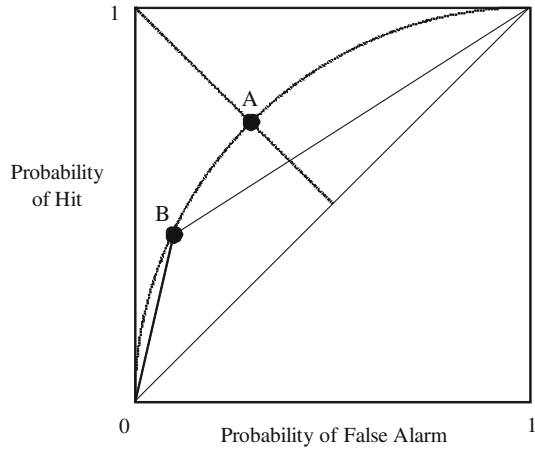
Signal detection theory is a formulation that has been widely used to assess human ability to detect signals (Green and Swets 1966). The premise of the paradigm is that there are two states of the world (signal vs. noise) and two possible human responses (I detect a signal vs. I do not detect a signal). The possible resulting states produces a 2×2 stimulus–response matrix shown in Fig. 2.5.

A key theoretical representation of signal detection theory is the receiver operating characteristic (ROC) (Swets 1996). The standard graphical depiction of the ROC is known as the ROC curve (Fig. 2.6). The curve reveals two important sources of information about operator performance: an individual’s decision criterion (the amount of evidence required to detect a signal); and the sensitivity of an individual’s detection performance (the individual’s ability to discriminate between signal and noise).

In order to apply SDT to the sensitivity analysis of time window outcomes, one must develop methods that do not violate assumptions of either formulation. In particular, the following three issues must be addressed: conversion of time window outcomes to SDT outcomes, calculating the probability of a false alarm in time window outcomes, and the development of a sensitivity measure without distribution assumptions.

The conversion of time window outcomes (Fig. 2.4) to SDT outcomes is dependent on a common definition of a hit. If a hit is defined to be an on-time and accurate action, so that Eq. 2.1 holds, then conversions from time window outcomes to SDT outcomes can readily be made. Table 2.1 shows the conversion

Fig. 2.6 The ROC curve under different distribution assumptions. If the distributions of signal and noise are normal, the sensitivity, d' , is determined by the distance of a point on the curve, point A, from the upper left diagonal. If no assumptions on the distributions can be made, the sensitivity can be approximated by the area under ROC (e.g., point B)



from time window outcomes to SDT outcomes. If an action is not executed on time, it is considered a false alarm. Therefore, a signal is only considered valid and detectable during a specified time interval in which the associated time window is designated open.

The original SDT formulation required forced-choice tasks primarily to ensure that correct rejections were accurate assessments of the absence of a signal. However, the decision environments for which time windows are intended are dynamic and interactive, and operators are not forced to take action. To calculate the probability of false alarm, which requires the number of false alarms and correct rejections, an accurate accounting method for correct rejections is needed. In fact, one method to measure correct rejections in these “free response” (Wickens and Kessel 1979) environments has already been developed. Wickens and Kessel (1979) computed the probability of false alarms as the number of false alarms divided by the number of false-alarm intervals. In their formulation, equal-valued intervals that span the detection task are separated into those that contain hits, and those that do not—called false-alarm intervals. Based on this concept, a false-alarm interval can be defined in the time window context. Consider the duration of a time window, T , over the lifetime of a simulation, T_s . The number of false-alarm intervals (FAI) can simply be formulated as:

$$FAI = \frac{T_s}{T} - 1 \quad (2.7)$$

The third issue to be addressed is the need for an appropriate sensitivity measure. If the distributions of the signal and noise are normal, the determination of the sensitivity, d' , can be visually determined from the ROC curve. In Fig. 2.6, for instance, the closer point A is from the upper left corner, the higher the sensitivity value. However, no assumptions can be readily made about distributions of signal and noise in dynamic domains. Therefore, one must rely on

Table 2.1 Conversion between time window outcomes and SDT outcomes

Time window outcome	SDT outcome
$ M^1(w_i, \mathbf{b}_j) _{T+,T-} = 1$	Hit
$ M^2(w_i, \mathbf{b}_j) _{T+,T-} = 1$	False alarm
$ M^3(w_i, \mathbf{b}_j) _{T+,T-} = 1$	
$ M^4(w_i, \mathbf{b}_j) _{T+,T-} = 1$	
$ M^5(\mathbf{b}_j) _{T+,T-} = 1$	
$ M^6(w_i) _{T+,T-} = 1$	Miss
Correct rejection	Correct rejection

nonparametric measures of sensitivity. Wickens and Hollands (2000) recommend a simple measure based on area under a ROC. The measure, first considered by Green and Swets (1966), is formulated as follows:

$$A_G = \frac{P(H) + [1 - P(FA)]}{2} \quad (2.8)$$

If only one point is acquired on the ROC, such as point B in Fig. 2.6, a sensitivity value can now be calculated. While these measures are still dependent on distributional assumptions (Caldeira 1980), they nevertheless serve as a good first approximation (Craig 1979).

The research methodology proposed here was implemented in a study to investigate tactical decision making performance under stress. For experiment details, see Rothrock (2001).

2.2 Analyses of Team Performance in a Dynamic Task Environment

In this part of the paper, team performance will be assessed from the perspective of time windows. Teamwork, a central component of team research, is not readily observable and must be inferred from the manner in which teams operate. Of particular interest is the measurement and evaluation of teamwork. The goal of this section is to explore the assessment of team data using a temporal accuracy measure called the relative accuracy index (RAI). The generalized mixed model will be used for the statistical analysis because of the type of data (binomial) and of the correlation structure within team members. The statistical procedure is described in detail to guide researchers who encounter similar problems. Using our statistical analysis, we found that participants whose training focused on coordination activities outperformed those whose training did not. Moreover, we found that workload stress accentuates the difference.

2.2.1 Introduction

An understanding of the relationship between team processes, outcomes and performance is a necessary prerequisite to the development of team training processes. Marks et al. (2002) argue that teamwork or team processes are the mediating links that link the relationship between team training and corresponding team outcome (performance) within the setting of input-process-outcome models. Coovert et al. (1990) suggest that team processes relate to the activities, strategies, responses, and behaviors employed in task accomplishment within teams. Team outcomes on the other hand pertain to the outcome of the various team processes. Any team performance measure or TPM (Cannon-Bowers and Salas 1997) must address the process as well as outcome measures in an appropriate manner.

Cannon-Bowers and Salas (1997) argue that TPMs must consider measurement at the individual and team levels because both teamwork and taskwork skills influence team performance. Additionally, TPMs must include measures that address process as well as outcome. The process measures describe the activities, strategies, responses and behaviors relevant to the human that are used to accomplish a task. In the past, researchers have used several instruments to assess and measure process and outcome measures for operator actions at both individual and team level. Smith-Jentsch and her colleagues (Smith-Jentsch, Johnston and Payne 1998) provide a list of such instruments including: sequenced actions and latencies index (SALI), behavioral observational booklet (BOB), anti-air teamwork performance index (ATPI) and anti-air teamwork observation measure (ATOM). While SALI and BOB are measures used to evaluate individual level outcomes and processes, ATPI and ATOM are used to evaluate team level outcomes and performance. These instruments are used by experts in the field to provide subjective ratings for process and outcome measures at individual and team levels, and provide an indication of the expert's judgment of operator performance. Therefore, these ratings are subject to problems such as inter-rater reliability. Additionally, the subjective ratings provided by the experts are often decoupled from the objective measures of team performance.

In contrast to the existing measures listed above, we focus on a measure called the RAI (Thiruvengada and Rothrock 2007). RAI circumvents the inter-rater bias problem as it does not involve expert ratings. It is an instrument that provides an objective assessment of process and outcome measures based on time windows. Given the potential time window outcomes, RAI can be expressed as the ratio of the number of 'on time' correct actions executed by an operator for a class of time windows to the total number of time windows that are opened in that class for that specific operator role. The mathematical formulation for RAI is shown in Eq. 2.9.

$$\text{RAI} = \frac{\text{Number of 'on time' correction actions for a class of time windows}}{\text{Total number of time windows that are opened in that class}} \quad (2.9)$$

In this study, team performance depends upon four teamwork dimensions: information exchange, communication, team initiative/leadership, and supporting behavior. The detailed explanation for each of these dimensions will be given later. Time windows that relate to a specific teamwork dimension, such as information exchange, are grouped together and are said to belong to the same class of time windows for calculating RAI.

2.2.2 Problem Domain

To demonstrate the utility of RAI, we conducted an empirical study with human participants using a human-in-the-loop simulation known as the *team Aegis simulation platform* (TASP). The objective of TASP is to reproduce a naval command-and-control environment in the combat information centre (CIC) task context (onboard a Navy ship with aircraft and missile launch capabilities) in which there are up to three operator roles functioning as a team, an anti air warfare coordinator (AAWC), an aircraft information coordinator (AIC) and a sensor operator (SO), acting concurrently. All operators have well defined tasks (responsibilities) set in a military context and are provided with rules of engagement (RoE) (Table 2.2) to help aid in their decision making process. The operators are recommended to follow the RoE at all times to achieve team goals. The RoE is different for each operator role in the team but governs their overall activities. Each operator is required to perform tasks based on RoE as well as compensate their teammates through supporting behavior (backup and error correction). The AAWC is the commander of team (team leader) and is responsible for coordinating the overall activities, including identifying unknown aircraft, assigning and engaging missiles on hostile aircrafts. The AIC is responsible for monitoring the activities of all friendly combat aircrafts, known as defensive counter air (DCA) and requesting visual identification (VID) report from them. The SO interprets any incoming sensor signals and issues warnings to hostile aircrafts violating the RoE.

There are several distinct as well as overlapping responsibilities among operator roles in TASP. At least one primary task responsibility on one role is shared among the other operator role, where the operator under whom the responsibility is listed has the primary action responsibility for that task. For example, the task of assigning primary identification label to any unknown aircraft is shared among the three roles, but the AAWC operator has the primary action responsibility for this task. Tasks on each role are executed through the use of a graphical user interface. As an example, Fig. 2.7 shows the graphical user interface for the AIC operator role.

The upper left box in Fig. 2.7 contains information about an object under consideration (e.g., an aircraft with an unknown identity). The spatial representation of objects in the vicinity of the AIC's ship is portrayed through the radar scope on the right half of the display. Action can be taken through the interface via function keys or buttons shown on the middle box in the left side of the display.

Table 2.2 Rules of engagement (RoE)

AIC	SO
1. Engage a Hostile aircraft within 20 nautical miles (NM) from ownship (hostile aircraft only). (AAWC RESPONSIBILITY BACKUP)	1. Issue level 3 warning to hostile aircraft only when it is within 20–30 nautical miles (NM).
2. Assign a missile to a hostile aircraft within 30 NM from ownship (hostile aircraft only). (AAWC RESPONSIBILITY BACKUP)	2. Issue level 2 warning to hostile aircraft only when it is within 30–40 NM.
3. Maintain safety of DCA (e.g., keep DCA away from danger zones of hostile aircraft, do not let DCA run out of fuel, etc.).	3. Issue level 1 warning to hostile aircraft only when it is within 40–50 NM.
4. Keep DCA within 256 NM from ownship.	4. Make a primary identification of air contact (i.e., friendly, hostile). ^a (AAWC RESPONSIBILITY BACKUP)
5. Keep DCA at least 20 NM away from ownship.	5. Evaluate, correlate and transmit all sensor value emissions that appear on the EWS interface.
6. Make a primary identification of air contact (i.e., friendly, hostile). ^a (AAWC RESPONSIBILITY BACKUP)	

^a Once an aircraft has come within 50 NM from ownship, it should be identified before it travels an excess of 10 NM. If an aircraft “pops up” within 50 NM it should be identified before it travels an excess of 10 NM

Two overarching rules

- (1) Defend ownship and ships in battle group
- (2) Do not engage friendly or civilian aircraft

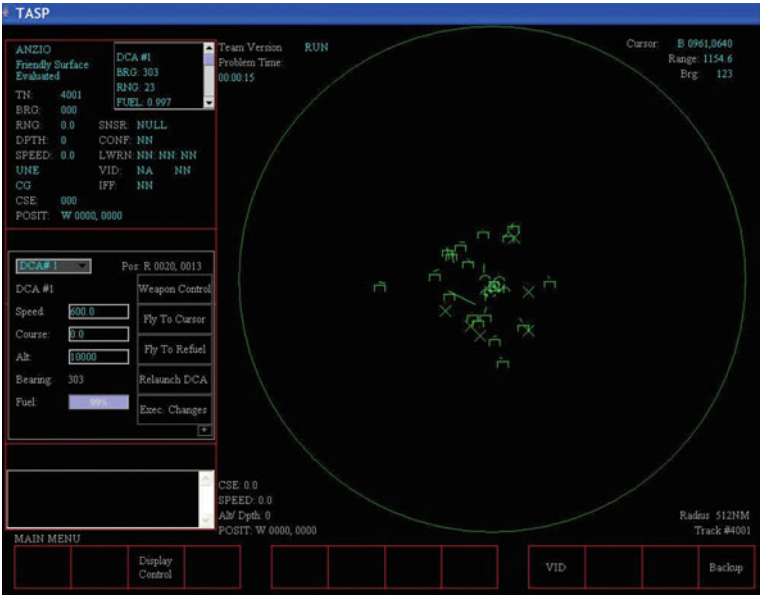


Fig. 2.7 Graphical user interface for an aircraft information coordinator (AIC) operator in TASP

For the purpose of this experiment, we consider a two person team of AIC and SO roles with no team leader AAWC. We use a simulation based approach that employs a truth maintenance system (TMS) in the background to keep tracks of information pertaining to time windows as well as operator actions. The time windows and operator actions data logged by TMS can be converted into a database using a converter tool in order to provide insights on metrics relating to RAI along teamwork dimensions. For example, when a hostile aircraft travels within 20 NM from the ownship, a time window is open specifying the opportunity of engaging the aircraft exists. When this aircraft travels out of the 20 NM range from ownship, the time window closes. If the AIC operator successfully engaged the aircraft within the time window, then AIC executed an ontime correct action. All these data are logged by TMS and the information can easily be queried from the database. The data can further be analyzed statistically to reveal the impact of the training intervention on team performance measures.

2.2.3 Teams and Performance Assessment Measures

Smith-Jentsch et al. (1998) defined four dimensions of teamwork for team dimensional training that are critical to overall team performance as information exchange, supporting behavior, communication and team leadership/initiative. Typically, these dimensions are assessed using post hoc surveys, questionnaires, and expert ratings. These dimensions are used to classify team outcome measures (time windows) into team process measures. While verbal communications existed between team members, specific content that was communicated between team members was not broken down to classify time window outcomes. Instead, time windows were opened and closed for each operator role based on the environmental conditions. These time windows are summarized in Table 2.3 and are classified into the teamwork dimensions.

The information exchange dimension relates to the process of gathering information and effectively exchanging them to develop a shared mental model for the team. Therefore, the AIC must fly a DCA within a certain distance to an unknown aircraft that approaches the vicinity of the ownship to gather its visual identification information (either friendly or hostile). In a similar fashion, the SO must detect sensor signal emissions and evaluate the intent of the signal as either friendly or hostile. The information that is gathered by both operators must be effectively exchanged among all team members. Therefore, time windows that are opened for visual identification and sensor evaluation process belong to this teamwork dimension. The tasks involved in this process are primary to the corresponding operator roles. Communication is external to the scope of this research as both operators exhibit implicit coordination without any overt communication. Both the AIC and SO operators must exhibit team initiative and leadership for the team's survival. Time windows pertaining to flying DCA out of potential threats and issuing warnings to approaching hostile aircrafts are classified under team

Table 2.3 Teamwork dimension classification of operator responsibilities for AIC and SO roles

Task type	Teamwork dimension	Responsibilities for operator roles	
		Aircraft information coordinator (AIC)	Sensor operator (SO)
Primary	Information exchange	Request visual identification (VID) report and pass it to other teammates	Evaluate incoming sensor signals
Backup	Communication	Operators did not use speech channels for communication (not considered)	Correlate sensor signal to a particular aircraft Transmit the correlated sensor signal Operators did not use speech channels for communication (not considered)
Primary	Team initiative/ leadership	Vector defensive counter air (DCA) within 256 NM from ownship Vector DCA outside 20 NM from ownship Vector DCA outside danger zones. (Vectoring of DCA is done by changing its speed, course and altitude)	Issue level one warning to hostile aircrafts Issue level two warning to hostile aircrafts Issue level three warning to hostile aircrafts
Backup	Supporting behavior	Assign identification to unknown aircrafts Assign missiles to hostile aircrafts Engage missiles upon hostile aircrafts.	Assign identification to unknown aircrafts
Error correct-ion		Change the identification of incorrectly identified aircrafts	Change the identification of incorrectly identified aircrafts

initiative/leadership dimension for AIC and SO, respectively. Finally, identifying the unknown aircraft and error correcting incorrect identifications are part of the supporting behavior dimension for AIC and SO roles. AIC is also responsible for supporting AAWC role by assigning and engaging a missile on hostile aircrafts that pose a high threat within close proximity to the ownship.

Consider the following example to translate behavior data into RAI outcomes. Suppose performance data is collected from a scenario as shown Table 2.4. Based on the classification of operator responsibilities along with teamwork dimensions (see Table 2.3), RAI scores can be calculated for each dimension. For example, AIC’s performance on the SB dimension involves two types of task activities, which are primary identifications (Primary ID) and assign and engage (A&E). We then can calculate RAI (AIC on SB) = [RAI (AIC on Primary ID) + RAI (AIC on A&E)]/

Table 2.4 Performance data from a sampled scenario

AIC								
Primary identification		VID		Assign and engage		RAI		
Opened time windows	On time correct actions	Opened time windows	On time correct actions	Opened time windows	On time correct actions	IE	TI/L	SB
16	2	2	2	1	0	01	NA	0.0625
SO								
Primary identification		Sensor operation		Issue level warnings		RAI		
Opened time windows	On time correct actions	Opened time windows	On Time correct actions	Opened time windows	On time correct	IE	TI/L	SB

$2 = (2/16 + 0/1)/2 = 0.0625$. In the same manner, the performance data for each group of participants can be collapsed to get their respective set of RAI scores.

2.2.4 Methodology

Participants in this research were first-year graduate and junior and senior-level undergraduate students at the Pennsylvania State University. A total of 78 students (39 two person teams), between the ages of 18 and 25, participated in this study. Of the total, 46 were male and 32 were female. They were skilled computer users and did not have any disabilities that restricted them from adequate use of mouse/keyboard interface. Additionally, the participants did not have any prior experience with the simulation environment. The participants engaged in a single session that lasted for about 3.5 h on average, and were provided with monetary compensation at the end of the study.

The two independent variables used in the study include training and workload. No training (NT), team coordination training (TCT) and task delegation training (TDT) are used as the training conditions. In NT condition, team members are not trained with any teamwork skills. They are required to read an article that illustrates the utility of team coordination and task delegation. However they are not provided with any information that prescribes how team coordination and task delegation can be achieved. Team members in the TCT condition are provided with excerpts of coordination strategies, which includes monitoring designated areas and passing information to other teammates as needed. The training helps team members in creating a shared mental model and allows them to anticipate the expectancies of their teammates. In TDT condition, specific tasks are delegated to the team members. The AIC is delegated tasks relating to assigning and engaging upon hostile aircrafts with missiles and issuing identifications based on visual identification information. The SO is delegated with tasks relating to assigning identifications based on sensors values that are evaluated. Differences between

Table 2.5 Types of training

No training (NT)	Team coordination training (TCT)	Task delegation training (TDT)
No specific training is imparted	Team coordination is emphasized during training	Task delegation is emphasized during training
Team members are provided with information on the definition of team coordination and task delegation	Team members are instructed on how to achieve effective coordination via demonstration of good and bad practices	The radar scope on the operator’s GUI is split into two distinct areas and is designated to each of the two roles. Operators monitor and perform actions within the designated area, while passing information pertaining to the other area onto their team mate
No specific tasks are delegated to each operator role	No specific tasks are delegated to each operator role	Specific tasks are delegated to each operator role based on KSA competencies and operator capabilities

training interventions are listed in Table 2.5. Workload stress levels are controlled by setting them at low and high levels. Different scenarios were developed for setting the stress levels of workload. The high stress workload scenarios include a relatively high number of hostile aircrafts that must be identified, assigned and engaged with missiles for both members within the team than in low stress workload scenarios.

Thirteen teams (one-third of 39 total teams) randomly received one of the three training conditions. The team members were randomly assigned to AIC or SO role. Each team was subjected to scenarios with both low and high workload stress levels.

The participants underwent an initial training of specific skills, which lasted for about an hour. This initial training enabled them to acquire skills that are necessary to accomplish tasks that are specific to their current roles. Four practice sessions (practice sessions 1–4) of 10 min duration each were provided to the participants to hone their role specific skills. During these practice sessions, the participants were given feedback on their performance relating to taskwork skills and were encouraged to ask any clarification questions. At the end of the four practice sessions, the teams were subjected to the first learning evaluation session for a duration of 10 min, which assessed their learning on taskwork skills. During this session, each team member was assigned specific tasks that would require them to use their taskwork skills and feedback about their performance was provided at the end of the session. After taskwork skills training, the teams were randomly exposed to one of the three team training interventions. In NT intervention, there was no hands-on training provided to the team regarding teamwork. Instead, they were instructed to read articles that explained the importance of teamwork and coordination. In “team coordination training” or TCT intervention, the teams were presented with instances of good and poor team coordination policies and were exposed to a video that demonstrated the

same. In TDT intervention, the teams were provided with a presentation of different tasks that were delegated to their roles as part of the training intervention and were also shown a video that demonstrated teamwork associated with task delegation. After the appropriate training intervention was provided, the teams were given an opportunity to practice teamwork skills through two 10 min practice sessions. Then, the teams were exposed to a second learning evaluation session that assessed their teamwork skills. The teams were instructed to perform tasks that required the effective use of taskwork and teamwork skills. The teams were then subjected to two sessions (of 10 min duration each) with low and high stress levels of workload where data relating to the performance of each team member (SO and AIC) were collected for further analysis.

2.2.4.1 The Statistical Model

The linear regression of team performance is modeled such that:

$$Y_i = \sum_{j=1}^J X_{ij}\beta_j + \varepsilon_i, \text{ where } \varepsilon_i \sim N(0, \sigma^2). \quad (2.10)$$

In such a model we assume that the error terms are normally distributed, zero mean and the same variances for all cases. However, outcomes that are proportions, as are the RAI's yield a distribution which violates the normality and homoscedasticity assumptions. Accordingly, analyzing proportions with linear regression may lead to misleading inference about the explanatory variables. This led researchers to consider logistic regression as the model for analyzing data in which the dependent variable is a proportion. The logistic regression is modeled as:

$$E(Y_i) = \mu_i = p_i = \frac{\exp(\sum_{j=1}^J X_{ij}\beta_j)}{1 + \exp(\sum_{j=1}^J X_{ij}\beta_j)}, \quad (2.11)$$

where, $E(Y_i) = p_i$.

Equation 2.11, that can also be expressed as:

$$\log \frac{p_i}{1 - p_i} = \sum_{j=1}^J X_{ij}\beta_j \quad (2.12)$$

is a particular case of the *Generalized Linear model*, in which linear regression models are extended to the exponential family of distributions that includes both the normal and the binomial distributions. Such models involve a link function which is some transformation $g(\cdot)$ that linearizes the expected value of Y_i , such that $g(\mu_i) = \eta_i$, and $\eta_i = \sum_{j=1}^J \beta_j X_{ij}$ is a linear combination of the predictors. The normal error regression model is a generalized linear model with the identity function as the link function, such that $\mu_i = \eta_i$. For logistic regression model

Table 2.6 Raw means and predicted means of the experimental data

Teamwork dimension	Training intervention						Workload stress			
	NT		TCT		TDT		Low stress		High stress	
	R ^a	P ^b	R	P	R	P	R	P	R	P
IE	0.2550	0.4556	0.4008	0.5631	0.3178	0.5039	0.3385	0.5545	0.2694	0.4607
TI/L	0.1714	0.1465	0.2434	0.2280	0.2823	0.2586	0.2697	0.2425	0.195	0.1795
SB	0.1549	0.1346	0.1629	0.1327	0.1740	0.1407	0.1704	0.1426	0.1574	0.1296

^a R is the raw mean from the observed data,

^b P is the predicted means by the model

Table 2.7 Type III test of fixed effects

Teamwork dimensions	Training intervention		Workload stress		Interaction	
	df	F	df	F	df	F
IE	(2, 36)	3.76 ^a	(1, 36)	13.94 ^b	–	–
TI/L	(2, 35.68)	1.05	(1, 72)	2.66	(2, 72)	4.48 ^a
SB	(2, 36)	0.81	(1, 36)	4.04 ^a	–	–

^a $p < 0.05$, ^b $p < 0.01$

$g(p) = \log \frac{p}{1-p}$, which is known as the logit function. Our experiment was designed to evaluate the effect of a certain type of training on an outcome Y, which is the proportion RAI. Since the dependent variable (RAI) is a proportion, the suitable distribution for modeling it, is the binomial distribution. The dependent variable Y in our experiment, was measured for each one of the two team members, at two stress levels (low/high), where each team belonged to one of three training groups (NT, TCT, TDT). The main aim in analyzing the data is to compare the groups on the outcome (RAI). For each of the 39 teams, divided randomly among the three types of training, there are four dependent measures of RAI since each team member (SO and AIC) has two outcome measures, corresponding to high and low levels of stress.

In the inference based on linear as well as generalized linear models, it is assumed that the observations are independent. Extending these models to account for correlated data led to the development of mixed models, for normal data, and more generally, to Generalized linear mixed models for the generalized linear models. Details of the model can be found in Rothrock et al. (2009).

2.2.4.2 Analysis and Results

The raw mean values and predicted mean values are shown in Table 2.6. Statistics of type III test of fixed effects are summarized in Table 2.7. The detailed analysis and results are elaborated for each teamwork dimension.

Table 2.8 Estimated covariance matrix for training and information exchange behavior

	AIC low	AIC high	SO low	SO high
AIC low	1.222 (0.285)	0.438 (0.233)	0.443 (0.283)	0.154 (0.403)
AIC high		1.286 (0.296)	0.311 (0.264)	−0.280 (0.428)
SO low			1.522 (0.382)	0.534 (0.356)
SO high				1.771 (0.590)

Standard errors are in parentheses

In the following, the standard errors (SE) of each estimate are displayed in brackets.

Training and Information Exchange

The analysis was performed based on 130 observations (26 were dropped due to zero value in the denominator). Since the interaction training × stress was found to be insignificant, it was dropped out from the model. The estimated covariance matrix for the experiment is shown in Table 2.8. From this matrix, we can observe the relationships of team member’s performance (AIC and SO) on different stress workload levels (low and high). Though not significant, we observe a negative correlation between the AIC and SO in the high stress condition. We also observe higher variances for SO, compared with the AIC.

The results indicate significant differences between the two training conditions TCT and NT ($p = 0.01$). The estimated RAI for TCT and NT were 0.563 (SE = 0.029) and 0.456 (SE = 0.028), respectively. Additionally, significant difference were found between the two stress levels ($p = 0.0007$), where the estimated RAI was 0.554 (SE = 0.017), for the low level of stress and 0.461 (SE = 0.025), for the high level.

Training and Supporting Behavior

The analysis was performed on 156 observations (no missing values).

Here too, the interaction training × stress was found to be insignificant, therefore it was dropped out from the model. The estimated covariance matrix for the experiment (Table 2.9) indicates negative and significant correlations between the AIC and SO both in the high and low stress conditions. In other words, when the RAI of the AIC was higher than average, the corresponding RAI of the SO was lower than average. A positive and significant correlation is observed between the low and high stress for each member. In other words, when a team member was higher/lower than average in one stress condition he was also higher/lower than average in the other stress condition. The results also indicate higher variances in the low stress condition, where the low stress variance of the SO was even higher than that of the AIC.

Table 2.9 Estimated covariance matrix for training and supporting behavior

	AIC low	AIC high	SO low	SO high
AIC low	3.058 (1.055)	2.012 (0.781)	−4.668 (1.205)	−1.449 (0.583)
AIC high		1.949 (0.702)	−2.835 (1.082)	−1.073 (0.358)
SO low			9.880 (2.683)	2.774 (1.254)
SO high				1.955 (0.666)

Standard errors are in parentheses

No significant difference was found among the training levels ($p = 0.81$), yet a significant difference was found between the two stress levels. The estimated RAI least-squares mean (lsmean) was 0.143 (SE = 0.006) for the low stress, while it was only 0.130 (SE = 0.007), for the high stress ($p = 0.05$).

Training and Team Initiative/Leadership

The time windows data indicated none of the AIC operators were able to perform the DCA manipulations in the experiment. Therefore we only have data corresponding to the SO (78 observations). Nevertheless, in order to allow a correlation between the two conditions measured for the same person, a repeated measures structure was used. The intra-class correlation, indicating the correlation within each team member (i.e., the correlation between two observations that belong to the same team member) was high (0.874).

For this outcome variable, the interaction between stress and training was significant, ($p = 0.013$). There are six different combinations of stress with training which led to 15 pairwise comparisons. Among these 15 tests, three were found to be significant. The most significant was the difference between the stress levels in the TCT training condition. The estimated RAI lsmean was 0.346 (SE = 0.093) for the low stress, and only 0.11 (SE = 0.039) for the high stress ($p = 0.002$). A significant difference was also found between the two training conditions TCT and TDT in the high stress condition ($p = 0.03$). While the estimated RAI lsmean was only 0.11 (SE = 0.039) for the TCT it was 0.288 (SE = 0.070) for TDT. Finally, a significant difference was also found between the NT group in the high stress and the TCT group in the low stress ($p = 0.04$), where the estimated RAI lsmean was 0.346 (SE = 0.093) for the low stress TCT and only 0.14 (SE = 0.046) for the high stress NT.

2.2.5 Discussion

The statistical analysis revealed an interesting view of team performance. Under the information exchange dimension—where information about the visual identity and sensor signature of tracks is shared—we found that TCT training significantly improved performance. Moreover, we also noticed a trend toward a negative

correlation between the AIC and the SO under stress, which suggests that teams tend to depend on a single source of information (either visual identification from the AIC or sensor information from the SO).

For the supporting behavior dimension, the effects of stress are more pronounced. A closer look at the type of activities involved with supporting behavior showed that they required longer key sequences to execute and that, under stress, fewer identification assignments were made. More importantly, as one role took on more activities under stress, the other role executed fewer activities. Therefore, just as information exchange tended toward uncertainty (i.e., only one source vs. two sources of information), supporting behavior also tended toward brittleness (i.e., one person assigning identities vs. two people).

In the team initiative and leadership dimension, our analysis discovered two interesting findings. The first is the absence of DCA activities, which suggests that the AIC either did not have the cognitive resources available to manipulate the DCA assets, or that the teams were not sufficiently trained to do so. In any case, the only data we had was the issuance of level warnings by the sensor operator. The second interesting finding was that participants exposed to TCT outperformed participants trained under either TDT or NT conditions. While the effect of the training was not universal across all stress combinations, our analysis suggests that TCT was more effective under high stress conditions. The comparison between the effects of TCT and TDT under the high stress condition was especially telling because TDT was developed to routinize responsibilities so that the effects of stress are mitigated.

2.3 Performance Assessment in an Interactive Call Center Simulation

In this part of the paper, a new performance assessment methodology call center systems at the level of customer-agent interactions (CAI) is proposed. A team-in-the-loop simulation test bed has been developed to analyze CAI-level performance using time windows. The proposed framework should allow researchers to collect and analyze individual as well as team performance at a finer granularity than current call center efforts.

2.3.1 Introduction

Today, we live in a service-based economy which faces challenges to assess and manage the performance of human-in-the-loop service systems (Chesbrough and Spohrer 2006). A case in point is the telephone call center which requires customer interactions for its operation. Because it is normally the first touch point of a business with which customers make contact, impressions on the total service

Table 2.10 Queue-centered call center measures

Measures	Description
Average speed of answer	The average time taken: for the call to be picked up
Average talk time	The average time that <u>acaller</u> waited to be connected to an agent
Queue time	The amount of time taken for a caller to wait in the line
Calls per hour	The average number of calls that an agent handles per hour
Hold time	The average time taken for an agent to place a customer on hold
Occupancy	The average time taken for an agent in his or her seat
Blocked calls	The total number of busy and out-of-order telephone trunks that block calls
Abandonment rate	The percentage of callers who disconnect prior to be answered
First call resolution	The percentage of calls closed on the first connect
Service level	Transactions that must be handled within given time frame

quality can be made from call center interactions. Traditionally, quality assessment has been made through direct call monitoring for every agent, which consumes tremendous amount of resources and times. In this regard, the proper modeling and evaluation of service systems can enable managers to effectively monitor service performance (Fleming et al. 2005).

Generally, a call center consists of trained customer service agents who answer customers' calls and coordinate their requests. Call center systems can provide a variety of functions such as help desk support, customer service, technical support, contact centers service, and tele-marketing etc. In this paper, we specifically focus on inbound call centers in which agents' assistance is sought by callers. Inbound call centers are very labor-intensive systems with high agent turnover rates amounting to "typically comprising 60–80% of the overall operating budget" (Aksin et al. 2007; Gilmore and Moreland 2000; Wallace et al. 2000). For this reason, managers tend to make an effort to improve the effectiveness of interaction between agents and customers through proper training and performance evaluation. Therefore, it should be priceless that managers can get a framework to provide quality information on their agents and customers interactions.

In previous research, the performance analysis of call centers has been mostly performed by using Erlang formulas that were designed for traditional queueing systems (Mehrotra and Fama 2003; Gilmore and Moreland 2000; Tanir and Booth 1999). These queueing based models may be useful and provide plentiful gross-level metrics in the case of evaluating the service performance in quantity assessments, as most call center research (Gans et al. 2003; Garnet et al. 2002) consider call centers as queueing system which consists of customers (callers), servers (telephone agents), and queues. Using this queue-centered approach, a variety of measures can be acquired and a representative sample of key performance indicators from Anton (1997) is shown on Table 2.10. Above all, the measure of a *telephone service factor* or *grade of service*, which is the percentage of calls answered in a given time frame, is widely used as a core measure (Sharp 2003). The previous works presented above, however, are only focused on quantity measures at a gross-level, while neglecting metrics of customer-agent interactions

which specify the service quality within an individual service activity of call center operations. Aksin et al. (2007) also noted that a macro research theme such as “improving the way in which the tension between efficiency and quality of service is modeled” is significant for future call center operations research. Therefore, one can no longer simply equate service quality with customer waiting times.

While queue-centered analytic models are still popular, Mehrotra and Fama (2003) noted that several factors such as complex call traffic, rapid change operations, and cheaper and faster computing, have recently increased the demand for analysis of ever more complex call centers through simulation. Although there are simulation approaches which deal with call center problems based on the optimization such as linear programming and scheduling (Avramidis et al. 2009; Atlason et al. 2004; Cezik and L’Ecuyer 2008), still they focus on gross-level metrics. However, in order to provide training feedback and manage call centers effectively with proper performance metrics, managers should know the quality of interactions between agents and customers during services.

To address the limitations of exiting analytic queue-centered approaches, this paper presents a configurable help desk call center team-in-the-loop (TITL) simulation test bed called the call center workforce simulation platform (CCWSP), which is the interactive simulation framework for performance analysis at the team as well as individual-task levels. The proposed framework uses time windows to develop a performance measure at the CAI level. Specifically, a new metric is proposed, called the index of interactive service performance (IISP), to measure service quality at CAI level with consideration of temporal service success rates within service operations. CAIs are expressed as pre-defined time windows and can be mapped to gross-level measures.

2.3.2 Human-in-the-Loop Discrete-Event Simulation

A human-in-the-loop simulation provides both realistic as well as controllable interactive task environments. With a human-in-the-loop simulation, users interact in real-time with the simulation through a graphical interface, and we can directly gather user data in a controlled experimental environment. In many service applications, however, agents in the systems may work as a team as well as individually. The team performance can be much more important than an individual performance when designing service operations with human inclusions.

2.3.3 The Proposed Framework: Call Center Workforce Simulation Platform with Time Windows

In this section, a framework for the CCWSP based on an interactive TITL simulation is presented. The Information Technology Services (ITS) help desk at Penn State is modeled as a problem domain of the simulation. The CCWSP software

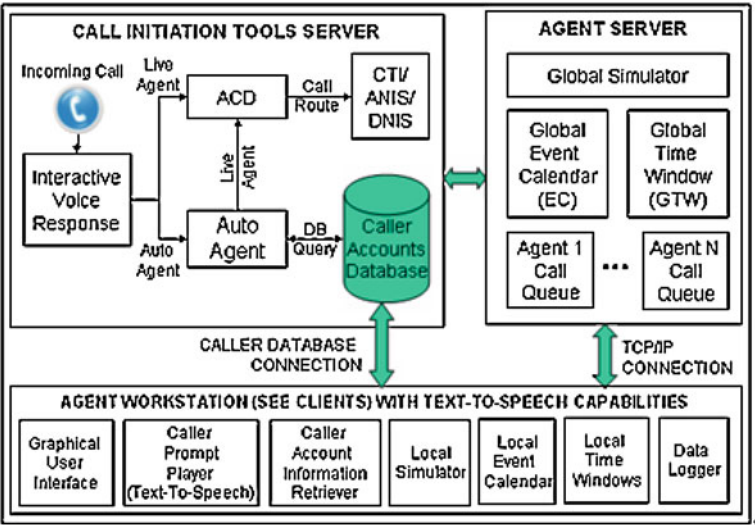


Fig. 2.8 Framework of the call center simulation application

architecture, the roles of time windows, and the call center simulation module are explained. In this discussion, we refer to the customer who makes a call to the call center as the caller and the service provider who answers a call as the agent.

The inbound help desk at Penn State University is used as the problem domain. The university ITS office runs the help desks system to handle calls by the Penn State user community on technical problems related to their computer hardware, software, network, and user account. In order to understand the help desk process, the operations of ITS help desk call center were analyzed through detailed field observations and a task analysis.

The application architecture for the help desk call center is built upon an interactive TITL discrete-event simulation that is comprised of three major parts: *Call Initiation Tools Server*, *Agent Server*, and *Agent Workstation* as shown in Fig. 2.8. The *Call initiation server* is a software component that provides an interface for live calls through computer telephony integration (CTI) equipment and updates the simulation about information pertaining to incoming calls. Currently, the server is driven by a pre-defined script file that simulates the caller information based on predefined scenarios. The *Agent Server* plays the role of a central server, not only in synchronizing the updates between various agents but also in placing a call in the caller queue as well as tracks gross-level performance metrics which can also be obtained from traditional queue-centered approaches. This server also maintains and tracks all individual workstation events through a global event calendar as well as windows of opportunity that exist for taking an action. The *Agent Workstation* simulates the events (based on a local event list and

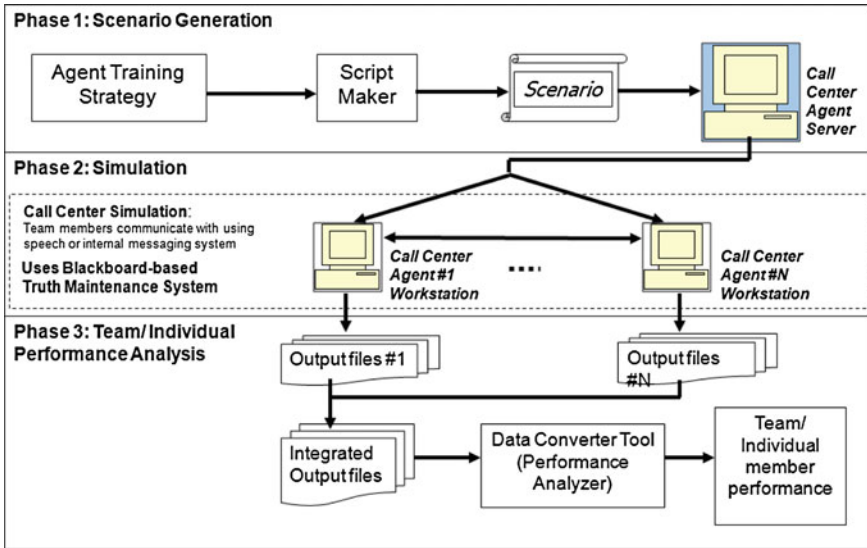


Fig. 2.9 Task flow diagram for the call center simulation framework

time windows) that are rendered on the graphical user interface (GUI) on each agent's workstation.

Three distinct phases collectively contribute to the performance measurement within the simulation framework. Figure 2.9 shows the task flow diagram of the three phases of the call center simulation framework. In Phase 1, the script files required to run the simulation are generated and initialized within the call center agent server. In Phase 2, the scenarios generated during Phase 1 are executed in real-time networked mode on each call center agent's workstation. This allows researchers to capture and log their actions into various data logs for further analysis. Finally, in Phase 3, the raw simulation output files are converted into a relational database for further performance evaluation.

In call center environments, the operations of time windows are not as restrictive and critical as those in command-and-control environments. Instead, a single operation is simply considered a link in the chain of the agent's activities required to perform a service call. For instance, if two consecutive actions (e.g., authentication and update a record) are needed to finish one service call, the situation of updating problems would be triggered by the agent's authentication action. On the other hand, in command-and-control operations, external factors based on rules of engagement, such as distance, altitude, and speed in a military radar system can situate agents' actions. Therefore, instead of specifying time duration for each time window, the opening and closing states are defined by agent's actions except in the case of a call drop. The latency in the agent's action is measured by the duration of each time window. As a result, only on time-correct actions, on time-incorrect actions, false alarms, and missed actions in Fig. 2.4 are

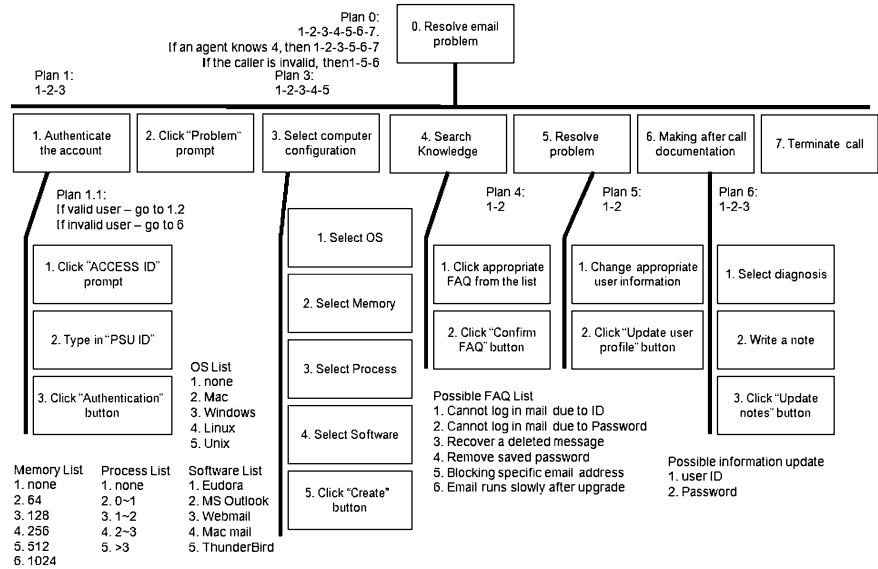


Fig. 2.10 HTA for a “Resolve email problem” task

possible results in the help desk simulation. The information of time window is logged for further performance analyses.

In call center environments, one call might be serviced by more than one agent (e.g., general agents can transfer technical calls to responsible agents). To handle this situation in a proper way, two levels of time windows are managed as global and local time windows. The *Agent Server* deals with global time windows to trace transferring calls, and the *Agent Workstation* takes care of each agent’s local time windows.

To demonstrate the multi-level time windows in call center environments, “Update PSU account” and “Resolve email problem,” are used as required tasks in our example domains. The result of hierarchical task analysis (HTA) of “resolve email problem” is illustrated in Fig. 2.10. This task is required to solve problems related to emails and has similar steps with the “updating PSU account” task except the “Selecting computer configuration.” Both tasks need to be authenticated by caller’s PSU ID. Once the caller’s PSU ID is valid, the agent can proceed to communicate to figure out the problem by clicking the “Problem” prompt. Or the agent can go to the step of making an after-call documentation directly and indicate the caller is invalid. Next, the use of knowledge base will be determined based on the level of agent’s expertness to provide proper information. By clicking the “Confirm FAQ” button, the agent determines whether the agent searches the right information. After changing the user’s profile and clicking the “Update user profile” button, an after-call documentation with a proper diagnosis should be made by clicking the “Update notes” button. Finally, the agent can terminate the

call. Within the two hierarchical tasks, 12 specific actions such as updating a user ID or removing a saved password are available.

Based on this task domain, the list of time windows is formulated as shown in Table 2.11. We define two types of time windows, primary and error-correction time window. The primary time window indicates the first available opportunity for an action. If the agent wants to revise his or her previous actions, then the error correction time window will gather the information. For the tasks at hand, there are a total of 19 time windows. Table 2.11 provides a breakdown of 11 primary and eight error-correction time windows with the action outcomes and open/close conditions.

Figure 2.11 illustrates the sequence of tasks along with the opening and closing conditions for each time window. At first, a caller places an incoming call which opens an overall time window for the call. Next, agents can see this incoming call on their call stack. Then, one of the agents picks up the call, which opens an authentication time window until the agent authenticates the caller. Once the caller is authenticated, the agent is able to complete other tasks such as creating problem profile, searching solutions, verifying user ID, verifying password, and resetting the password information. The TMS (truth maintenance system) opens primary time windows for these processes until the related action is performed. For example, if the agent creates a problem profile, the primary problem time window is closed by TMS and a secondary error correction time window is opened and remains open until the call ends. When the error correction time window is opened, the agent can correct any previous incorrect actions and all such agent actions along with the time window information are recorded in the output files for further performance evaluation.

From the time window's structure, we can categorize agent performances. If a time window is opened but no related agent action exists, then such an action is treated as a *missed action*, as shown in Fig. 2.9. On the other hand, if a time window is not opened but an agent action exists, then such actions would be related to a *false alarm action*. Only a related action is taken when a time window is opened can the action be considered on time and correct.

After gathering the time window information, the agent's performance at the CAI level is evaluated during the data analysis phase. In comparison to the queue-centered measure which gives overall values of system performances, the time windows-based measure in call center systems would give more detailed performance information related with human-interactions and deeper insights for call center managers. Nonetheless, the *Agent Server* in this simulation framework provides queue-centered measures, too.

To analyze and evaluate the quality of services in the CAI level, the appropriate method to describe quantitative service-related parameters is necessary. In the existing time windows approaches, Rothrock (2001) provide two method of evaluating operator performance based on time windows. The first method of factor analysis represents a technique that reduces factors to evaluate which situations and operator actions can be aggregated into higher order factors. The second method, signal detection theory (SDT), is designed to reveal an individual's decision criterion and the sensitivity of an individual's detection performance. Based on these

Table 2.11 List of 19 time windows in the task domain

Name of time window	Type	Open condition	Close condition
Overall call	<i>Primary</i>	Calls comes in the simulation	Call drop or end
Authentication	<i>Primary</i>	Pick-up button clicked	Authentication button clicked or call drop
Computer profile	<i>Primary</i>	Authentication button clicked	Create profile button clicked or call drop
Error correction for computer profile	<i>Error correction</i>	Create profile button clicked	Call drop or end
Diagnosis for the problem	<i>Primary</i>	Authentication button clicked	Confirm button in knowledge base clicked or call drop
Error correction for diagnosis	<i>Error correction</i>	Confirm button in knowledge base clicked	Call drop or end
Change password	<i>Primary</i>	Authentication button clicked under proper problem ID	Updated in password text field and reset button clicked
Reset ID	<i>Primary</i>	Authentication button clicked under proper problem ID	Updated in user ID and reset button clicked
Change name	<i>Primary</i>	Authentication button clicked under proper problem ID	Updated in name test field and reset button clicked
Change phone number	<i>Primary</i>	Authentication button clicked under proper problem ID	Updated in phone number text field and reset button clicked
Change E-mail address	<i>Primary</i>	Authentication button clicked under proper problem ID	Updated in e-mail address text field and reset button clicked
Change address	<i>Primary</i>	Authentication button clicked under proper problem ID	Updated in address text field and reset button clicked
Error correction for Change password	<i>Error correction</i>	Updated in password text field and reset button clicked	Call drop or end
Error correction for reset ID	<i>Error correction</i>	Updated in User ID and reset button clicked	Call drop or end
Error correction for change name	<i>Error correction</i>	Updated in name text field and reset button clicked	Call drop or end
Error correction for change phone number	<i>Error correction</i>	Updated in phone number text field and reset button clicked	Call drop or end
Error correction or change E-mail address	<i>Error correction</i>	Updated in e-mail text field a and reset button clicked	Call drop or end
Error correction for change address	<i>Error correction</i>	Updated in address text field and reset button clicked	Call drop or end
Notes	<i>Primary</i>	Authentication button clicked	Update button clicked or call drop

two methods, Thiruvengada and Rothrock (2007) suggested the RAI to evaluate team performance in a command-and-control human-in-the-loop simulation. The RAI can give quick and quantitative measures of performance data for system evaluation.

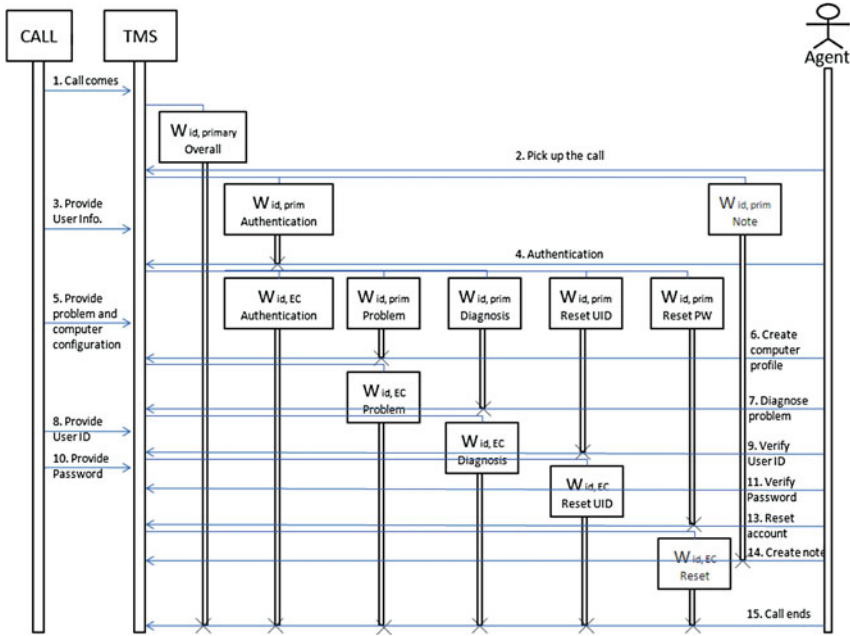


Fig. 2.11 Sequence diagram for time windows

However, the proposed index of RAI is difficult to be applied to measuring the CAI level metrics in service systems because of its strict adherence to time constraints in fixed-rule domain such as military operations.

To make the quantitative measures and analyze time windows-based measures in call centers by linking up with service quality and customer satisfaction, we suggest a new index, called an Index of IISP. We define IISP as an agent's ability to provide the correct service within a service level. The term service level refers to transactions that must be handled on arrival at the call centers. In this paper, the service level corresponds to the one of service quality standards and is expressed as the time limitation of the service. IISP is interpreted as the ratio of the number of "within service level" correct actions conducted by an agent for a class of time windows to the total number of time windows that should be opened in the class. Because there are two types of time windows, primary and error correction, in the list of time windows in Table 2.2, total time windows would be calculated as an average of the two types of time windows. The mathematical representation for IISP is formulated as follows, where $SL_Correct$ is the area ① in Fig. 2.12:

$$IISP = \frac{\sum_{i=1}^n SL_Correct(i)}{n} \quad (2.13)$$

(SL = service level, n = number of time windows)

	Situation required		No Situation required
Action		(Within) SL	FA ④
	Correct	①	
	Incorrect		
No Action	Miss ③		CR

Fig. 2.12 Possible time window outcomes

Table 2.12 Metrics of human performance based on time windows

	IISP	RAI	Factor analysis	SDT
Purpose	Quantitative index for time windows in less strict domain (service domain)	Quantitative index for time windows in fixed-rule domain (command-and-control)	Determination of correlations among different time windows	Determination of the sensitivity of operator actions to situation requirements
Outcome	The ratio of the time windows number of “within service level” correct actions to the total number of time windows that should be opened	The ratio of the time windows number of ontime-correct actions to the total number of time windows that should be opened	Correlations among factors	ROC curve which is represent the receiver operating characteristic
Note	A guideline of IISP score is needed for subject matter experts	A guideline of RAI score is needed for subject matter experts	Screen test can be used	Accurate accounting method for correct rejections is needed

The comparison among the performance measures of time windows is shown in Table 2.12. The proposed IISP is for less strict domains such as service systems and can give users to judge the service quality and performance of the systems. For more details on a simulation study involving CCWSP, see Ma et al. (2011).

2.3.4 Discussion

The IISPs can show the degradation of service qualities while queue-centered gross level approach, which counts only on processing time, cannot capture the overall service performance in detailed levels. Time windows approach provides information of what kinds of specific sub-processes are required to be improved for either an individual agent or a group of agents.

The proposed framework consists of not only queue-centered measures but also CAI ones. In particular, small-sized call centers could benefit from IISP measures due to the large variance of agents' performance. For middle and large-sized call centers, the framework also provides benefits in terms of training and investigating agents' performance under interested situations. If the agents repeatedly miss or fail some time windows, then remediation can be the training of tasks that improve performance on those windows. Also, the framework enables managers to simulate specific situations or new service systems. For example, if a manager wants to know the effect of new call distribution system towards agents' performance, then he or she can compare the simulation results in the system.

In comparison with gross measures such as queue time and call duration (talk time), IISP is more diagnostic of individual tasks performed. Therefore, managers can easily understand both a system and workforce information with it. The detailed meaning of IISP would be captured from the raw time windows information. IISP also enables managers to compare their agents and help to generate a workforce performances profile.

In order to analyze time windows information from CCWSP, time windows must be categorized and defined clearly. For the ITS help desk task domain, 19 time windows were pre-defined. Also, managers need to set the service level correctly. Finally, by testing participants with the target scenario, time windows information can be gathered, and IISP can be calculated along with other queue-centered measures.

2.3.5 Conclusions

A research approach to evaluate operator performance in human-in-the-loop simulations has been proposed. The key concept within the approach is a notion of time windows. The time window construct provides a computational framework to dynamically evaluate operator actions in the context of heterogeneous task demands.

To implement time windows in a working model, a blackboard paradigm was introduced. The blackboard model is suited to accommodate the time window construct because of its ability to reason opportunistically about the availability of situations and the timeliness of operator actions. It was argued that human-in-the-loop simulations are ideal tools to investigate dynamic phenomena without concerns of the oversimplified laboratory environment or the unconstrained real-world. Therefore, requirements for implementation of the blackboard model were discussed. Moreover, a study which implemented the blackboard model in a human-in-the-loop simulation was used to illustrate the viability of the time window construct to provide a framework for operator performance. Two methods for analysis of time window outcomes were discussed to provide complementary perspectives on operator attunement to the constraints.

Time windows were then used to develop the RAI as a measure of team performance, a proposed standard that cuts across disciplines and enables the use of statistical techniques to aid researchers in better understanding team decision

making. By using RAI as the primary metric, inter- and intra-rater reliability difficulties faced by the researchers are avoided. Ultimately, the effectiveness of an RAI-based measure is contingent on the ability of evaluators to establish the rules that govern a particular task domain. For example, temporal rules in a command-and-control domain are fairly straight-forward to extract whereas rules in a political debate are much more difficult to obtain. In general, RAI-based measures are more effective in domains where standard operating procedures and time constraints are clearly defined.

Finally, RAI was extended to a service enterprise—the call center. An approach using time windows-based assessment of an inbound call center system was proposed, which enables researchers not only to explain queue-centered measures utilized by most call center researchers, but also to explicate CAI measures. A configurable Team-in-the-loop simulation of a help desk, the CCWSP, was used to demonstrate the utility of this methodology. We also suggested a new quantitative index of agent performance, the Index of IISP which can provide a quantitative analysis of the agent service performance based on time windows. From the IISP, time windows-based measures from CCWSP can be systematically analyzed.

References

- Adeli H, Yu G (1995) An integrated computing environment for solution of complex engineering problems using the object-oriented programming paradigm and a blackboard architecture. *Comput Struct* 54(2):255–265
- Aksin O, Armony M, Mehrotra V (2007) The modern call-center: a multi-disciplinary perspective on operations management research. *Prod Operat Manag* 16(6):655–688
- Allen JF (1983) Maintaining Knowledge about Temporal Intervals. *Commun ACM* 26(11): 832–843
- Anton A (1997) *Call center management: by the numbers*. Purdue University Press, West Lafayette
- Atlason J, Epelman MA, Henderson SG (2004) Call center staffing with simulation and cutting plane methods. *Ann Oper Res* 127:333–358
- Avramidis A, Chan W, L’Ecuyer P (2009) Staffing multi-skill call centers via search methods and a performance approximation. *IIE Trans* 41(6):483–497
- Beach LR, Lipshitz R (1993) Why classical decision theory is an inappropriate standard for evaluating and aiding most human decision making. In: Klein GA, Orasanu J, Calderwood R, Zsombok C (eds) *Decision making in action: models and methods*. Ablex Publishing Corp, Norwood, pp 21–35
- Booch G, Rumbaugh J, Jacobson I (1999) *The unified modeling language user guide*. Addison-Wesley, Reading
- Caldeira JD (1980) Parametric assumptions of some “nonparametric” measures of sensory efficiency. *Hum Factors* 22(1):119–120
- Cannon-Bowers JA, Salas E (1997) A framework for developing team performance measures in training. In: Brannick M, Salas E, Prince C (eds) *Team performance assessment and measurement: theory, methods, and applications*. Erlbaum, Hillsdale, pp 45–62
- Cattell RB (1966) The meaning and strategic use of factor analysis. In: Cattell RB (ed) *Handbook of multivariate experimental psychology*. Rand McNally, Chicago

- Cezik MT, L'Ecuyer P (2008) Staffing Multiskill Call Centers via Linear Programming and Simulation. *Manag Sci* 54(2):310–323
- Chesbrough H, Spohrer J (2006) A research manifesto for services science. *Commun ACM* 49(7):35–40
- Chu RW, Jones PM, Mitchell CM (1991) The Georgia tech payload operations control center simulation: design and implementation in C++. In: *Proceedings of the object-oriented simulation track of the 1991 society for computer simulation multicongress*, Anaheim, CA
- Coovert MD, Cannon-Bowers JA, Salas E (1990) Applying mathematical modeling technology to the study of team training and performance. Paper presented at the 12th annual interservice/industry training systems conference
- Craig A (1979) Nonparametric measures of sensory efficiency for sustained monitoring tasks. *Hum Factors* 21(1):69–78
- Endsley MR (2006) Situation awareness. In: Salvendy G (ed) *Handbook of human factors and ergonomics*, 3rd edn. John Wiley and Sons, Inc, New York, pp 528–542
- Ermann DL, Hayes-Roth F, Lesser VR, Reddy D (1980) The HEARSAY-II speech understanding system: integrating knowledge to resolve uncertainty. *ACM Comput Surv* 12:213–253
- Fleming JH, Coffman C, Harter JK (2005) Manage your human sigma. *Harv Bus Rev* 83(7/8):106–114
- Gabbay DM, Hodkinson I, Reynolds M (1994) *Temporal logic: mathematical foundations and computational aspects*. Oxford University Press, Oxford
- Gans N, Koole G, Mandelbaum A (2003) Telephone call centers: Tutorial, review, and research prospects. *Manuf Serv Oper Manag* 5(2):79–142
- Garnet O, Mandelbaum A, Reiman M (2002) Designing a call center with impatient customers. *Manuf Serv Oper Manag* 4(3):208–227
- Gilmore A, Moreland L (2000) Call centres: how can service quality be managed? *Ir Market Rev* 13(1):3–11
- Green PE (1978) *Analyzing multivariate data*. Dryden Press, Hinsdale
- Green DM, Swets JA (1966) *Signal detection theory and psychophysics*. Wiley, New York
- Greeno JG (1994) Gibson's affordances. *Psychol Rev* 101(2):336–342
- Greeno JG (1998) The situativity of knowing, learning, and research. *Am Psychol* 53(1):5–26
- Greeno JG, Moore JL (1993) Situativity and symbols: response to Vera and Simon. *Cogn Sci* 17:49–60
- Hammond KR (1986) Generalization in operational contexts: what does it mean? can it be done? *IEEE Trans Syst Man Cybern*, SMC 16(3):428–433
- Howie DE, Vicente KJ (1998) Measures of operator performance in complex, dynamic microworlds: advancing the state of the art. *Ergonomics* 41(4):485–500
- Jones PM, Chu RW, Mitchell CM (1995) A methodology for human-machine systems research: knowledge engineering, modeling, and simulation. *IEEE Trans Syst Man Cybern* 25(7)
- Kirlik A (1995) Requirements for psychological models to support design: toward ecological task analysis. In: Flach JM, Hancock P, Caird J, Vicente KJ (eds) *Global perspectives on the ecology of human-machine systems*. Erlbaum, Hillsdale, pp 68–120
- Laudeman IV, Palmer EA (1995) The quantitative measurement of observed workload in the analysis of aircrew performance. *Int J Aviat Psychol* 5(2):187–197
- Ma J, Kim N, Rothrock L (2011) Performance assessment in an interactive call center workforce simulation. *Simul Model Pract Theory* 19:227–238
- Marks MA, Sabella MJ, Burke CS, Zaccaro SJ (2002) The impact of cross-training on team effectiveness. *J Appl Psychol* 87(1):3–13
- Mehrotra V, Fama J (2003) Call center simulation modeling: methods, challenges, and opportunities. Paper presented at the 2003 winter simulation conference
- Militello LG, Hutton RJB (1998) Applied cognitive task analysis (acta): a practitioner's toolkit for understanding task demands. *Ergonomics* 41(11):1618–1641
- Moray N, Dessouky MI, Kijowski BA, Adapathya R (1991) Strategic behavior, workload, and performance in task scheduling. *Hum Factors* 33(6):607–629

- Nii HP (1986) Blackboard systems: the blackboard model of problem solving and the evolution of blackboard architectures. Part I. *AI Mag* 7:38–53
- Raby M, Wickens CD (1994) Strategic workload management and decision biases in aviation. *Int J Aviat Psychol* 4(3):211–240
- Rothrock L (2001) Using time windows to evaluate operator performance. *Int J Cogn Ergonomics* 5(1):1–21
- Rothrock L, Cohen A, Yin J, Thiruvengada H, Nahum-Shani I (2009) Analyses of team performance in a dynamic task environment. *Appl Ergonomics* 40(4):699–706
- Rubin KS, Jones PM, Mitchell CM (1988) OFMspert: inference of operator intentions in supervisory control using a blackboard architecture. *IEEE Trans Syst Man Cybern SMC* 18(4):618–637
- Sanderson PM, Verhage AG, Fuld RB (1989) State-space and verbal protocol methods for studying the human operator in process control. *Ergonomics* 32:1343–1372
- Sharp D (2003) Call center operation: design. Digital Science Operation, and Maintenance, New York
- Smith-Jentsch KA, Johnston JH, Payne SC (1998) Measuring team-related expertise in complex environments. In: Cannon-Bowers JA, Salas E (eds) *Making decisions under stress: implications for individual and team training*. APA, Washington, D.C., pp. 61–87
- Smith-Jentsch KA, Zeisig RL, Acton B, McPherson JA (1998) Team dimensional training: a strategy for guided team self-correction. In: Cannon-Bowers JA, Salas E (eds) *Making decisions under stress: implications for individual and team training*. APA, Washington, D.C., pp 271–297
- Suchman LA (1987) *Plans and situated actions: the problem of human machine communication*. Cambridge University Press, Cambridge
- Swets JA (1996) Signal detection theory and ROC analysis in psychology and diagnostics: collected papers. Lawrence Erlbaum Assoc, Mahwah
- Tanir O, Booth RJ (1999) Call center simulation in Bell Canada. Paper presented at the 1999 winter simulation conference. Phoenix, AZ
- Thiruvengada H, Rothrock L (2007) Time window-based performance measures: a framework to measure team performance in dynamic environments. *Cogn Technol Work* 9(2):99–108
- Vicente KJ, Rasmussen JR (1992) Ecological interface design: theoretical foundations. *IEEE Trans Syst, Man Cybern* 22(4):589–606
- Vranes S, Lucin M, Stanojevic M, Stevanovic V, Subasic P (1991) Blackboard metaphor in tactical decision making. *Eur J Oper Res* 61:86–97
- Wallace CM, Eagleson G, Waldersee R (2000) The sacrificial HR strategy in call centres. *Int J Serv Ind Manag* 11(2):174–184
- Wickens CD, Hollands JG (2000) *Engineering psychology and human performance*, 3rd edn. Prentice Hall, Upper Saddle River
- Wickens CD, Kessel C (1979) The effects of participatory mode and task workload on the detection of dynamic system failures. *IEEE Trans Syst Man Cybern* 9(1):24–34



<http://www.springer.com/978-0-85729-882-9>

Human-in-the-Loop Simulations

Methods and Practice

Rothrock, L.; Narayanan, S. (Eds.)

2011, VIII, 261 p., Hardcover

ISBN: 978-0-85729-882-9