INTERNATIONAL JOURNAL OF HUMAN–COMPUTER INTERACTION, *16*(2), 235–260


# *A Systemic-Structural Activity Approach to the Design of Human–Computer Interaction Tasks*

**Gregory Bedny**
Division of Social Sciences
Essex County College

**Waldemar Karwowski**
Center for Industrial Ergonomics, Department of Industrial Engineering
University of Louisville

In this article, a new approach to the study of human–computer interaction (HCI) from the activity theory perspective is presented. A computer-based task was selected for demonstration purposes. Due to its complexity, variability, and number of mental components, the selected computer-based task presented difficulties in observation and formal description. Other tasks involving computers bared similar difficulties. In this study, it is demonstrated that activity theory, which has precise units of analysis and carefully elaborated concepts and terminology, can be useful in the study of HCIs. The examination and description of the computer-based task in this study are carried out through a systemic-structural analysis approach associated with activity theory.

## *1. INTRODUCTION*

At present, there is a gap between research studies and applied human–computer interaction (HCI) design. Some scientists assert that software designers can derive the relation between theoretical studies and practical guidelines from the human information approach alone. More commonly, guidelines are based on practical experience without the use of an underlying theory. As a result, some of the design recommendations are fragmented and may lack precision or coherence (for further discussion, see Bannon & Bodker, 1991; Kuutti, 1996, Nardi, 1997). Despite such deficiencies, cognitive psychology that is based on human information processing may be adapted for practical use. The human information approach was developed as a theoretical basis for the study of the psychological aspects of human brain functions. Frequently, supplementary methods of study develop theoretical concepts enabling the application of theories to practice.

Requests for reprints should be sent to Waldemar Karwowski, University of Louisville, Department of Industrial Engineering, Lutz Hall, Room 445, Louisville, KY 40292. E-mail: karwowski@louisville.edu

The concept of action provides a way to transcend the limitations of prevailing mentalist notions and bridge the practical domain. Through actions, participants alter their external world while simultaneously forming their mental models (Rubinshtein, 1973). Mental and behavioral actions and operations are major units of analysis in activity theory, rendering this theory decisive in the study of human work. Human cognition is organized as a continuous process (Brushlinsky, 1979). At the same time, this process is organized into a hierarchy of recursive subsystems directed to achieve goals of various operations and actions (Bedny, Seglin, & Meister, 2000). Hence, cognition should be studied as a continuous processing system and as a system of cognitive actions and operations. From a theoretical and practical perspective, the cognitive approach should be integrated with an activity approach. In our concept of the systemic-structural analysis and design of activity, cognitive analysis is one stage in the analysis of human performance (Bedny & Karwowski, 2000; Bedny, Karwowski, & Kwon, 2001).

### 1.1. Objectives

The main objective of this article was to explore the usefulness of the systemic-structural theory of activity (SSTA) to the study of HCIs. SSTA builds on the general theory of activity that originated in the former Soviet Union and now is widely recognized worldwide. The founders of the general activity theory were Rubinshtein (1963), Leont'ev (1977), and Vygotsky (1978). In Europe, the early work of Volpert (1982), Hacker (1985), and others (e.g., see Nardi, 1996; Engestrom, Miettinen, & Punamaki, 1999) influenced this field. However, in the past, the general activity theory was not sufficiently adapted to the study of human performance at work.

The last decade witnessed development of the SSTA that is specifically tailored to the study of work (Bedny, 2000; Bedny, Karwowski, & Bedny, 2001; Bedny, Karwowski, & Kwon, 2001; Bedny & Meister, 1997, 1999; Bedny & Seglin, 1999). Within SSTA, one can isolate three different approaches: (a) the parametrical approach, which concentrated on the study of different parameters of activity such as cognitive analyses, errors analyses, and so forth; (b) the morphological approach in which mental and motor actions concepts are the most important ones; and (c) the functional approach in which the major concepts are self-regulation and function blocks. From the SSTA perspective, task analysis focuses on the structure of activity during the task performance. Activity is presented as a system comprised of discrete hierarchically organized elements that are units of analysis. The content of these elements and the specifics of their interaction characterize the structure of activity. A description of the structure of activity as a system is called the systemic-structural analysis and description of activity.

With some exceptions (see Decortis, 2000; Kuutti, 1996), most of the current approaches in the HCI area typically concentrate on the systematic description of the human–computer interface but not on human performance itself as a system. Systemic-structural analysis is possible only because activity theory has developed units of analysis and corresponding methods to guide it. Actions, operations, function blocks, and members of an algorithm of activity supply the conceptual framework for such analyses. Systemic-structural analysis of activity employs distinct methods that are categorized in terms of specific criteria. Activity as an object of study embodies multiple, distinct aspects enabling the systemic-structural approach to capture the multidimensionality of activity during task performance. Therefore, multiple approaches must be employed to describe a single episode of activity.

The systemic-structural approach allows one to formulate activity as a multidimensional system and to describe those diverse models that capture various aspects of its structure. The method of systemic-structural analysis of activity suggests four stages: (a) qualitative description, (b) algorithmic analysis, (c) time structure analysis and (d) quantitative analysis. The major importance of the latter is the objective evaluation of task complexity. All stages are related to one another in a recursive, loop structure. Later stages frequently require revisiting preliminary stages of analysis. At any stage of the design process, activity may be described at different levels of detail or decomposition beginning with a gross description and followed by a more detailed one. Each level of analysis also has a recursive loop structure (Bedny, Karwowski, & Jeng, 2000; Bedny, Karwowski, & Kwon, 2001). Depending on the purpose of a study and the specificity of the object of study, some stages of analysis and description may be abbreviated or eliminated altogether.

### *1.2. Morphological Analysis of Task Performance*

In this article, we delineate a morphological analysis of task performance as a major method of study. *Morphological analysis* refers to the division of activity into actions and operations, algorithmic descriptions of activity, and developing time structure of activity. During the algorithmic description of activity, actions and operations become major units for the definition of members of the algorithms. The method for isolating actions in a holistic structure of activity is described in Bedny, Karwowski, & Kwon (2001). In turn, members of the algorithm are the basic units of an algorithmic description of activity. Usually, a member of an algorithm consists of one or several mental or motor actions integrated by the general goal. Algorithmic description serves as a basis for a practical application, which we consider in greater detail in what follows. On the basis of algorithmic description, activity is delineated in terms of a logically organized system of actions and operations distributed in space and time. If necessary, each element of activity may be treated as a subsystem entailing further decomposition.

As a practical case, in this study a task within an inventory process system of a manufacturing firm was selected. This inventory process includes receiving parts, putting them away, storage, withdrawal, and movement through work-in-process (WIP) while simultaneously tracking their movement as well as recording all of these events. In this study, we focused on the receiving task. This is a computer-based task in which an operator performs his or her job mostly through computer interactions. We conducted the study within a framework of systemic-structural analysis of work activity. This study of computer-based task performance is limited to some aspects of qualitative analysis, which are restricted to parametric methods of analysis and algorithmic description of activity, which are related to morphological analysis.

## 2. QUALITATIVE STAGE OF ACTIVITY ANALYSIS

This stage of analysis can be performed based on parametric or functional analysis of activity. In simple cases, when general qualitative analysis of work performance is discussed, scientists use a parametrical method of study that allows concentration on distinct aspects of activity, as was carried out in this study.

Qualitative analysis of activity starts with objectively logical analysis. It may be reduced to providing a short verbal description of job performance, analysis of related production operations or tasks, and determination of their logical organization in space and time. Furthermore, a short description of technological processes is performed including description of major equipment, tools, raw materials, sequence of basic technological procedures, and so forth. Conditions of work such as temperatures, noise, illumination, and so forth also should be described, including the potential for extreme situations. The relation between computerized and noncomputerized components of work and the calculated proportion of time in computer-based work are called for.

Activity theory is distinctive in the attention it dedicates to sociocultural aspects of activity and the role of external and internal tools of activity in task performance (Vygotsky, 1978). Culture is considered as the mediator between the user and technology. It is the aggregation of beliefs, attitudes, values, social norms, and standards. Culture is comprised of shared social meanings. Individuals affiliated with a culture internalize these shared meanings. The social context under which a task is performed should be noted, including the social dynamics of the group involved in job performance. All of these questions may be addressed in broad or detailed fashion.

Another aspect of activity analysis is individual-psychological study. Personal requirements for job performance such as individual features of personality, educational background, motivational aspects of work, needs and desires, and wishes should be considered at this stage of analysis. The background and training of users in the use of a computer and their subjective relation to computerization are analyzed along with relative job satisfaction from the use of a computer.

During the next stage, researchers perform detailed task analysis or analysis of production operations in which they attend to separate tasks or production operations. Specialists become involved in a detailed description of task performance. Description of the structure of activity becomes important at this stage. This method of analysis may be transformed into a cognitive or functional analysis of task performance with which it is intimately connected. In some cases, qualitative methods of analysis suggest the use or development of symbolic process models. These models are distinctive insofar as each symbol refers to a unit of analysis in activity theory. Depending on the degree of detail, this analysis may be at either a microstructural or macrostructural level (Bedny, 2000). During the qualitative analysis of activity, widely used methods of comparative analysis, such as contrasts between effective and substandard performers, worker's error analysis, and definition of difficulties and obstacles, are used. The workers' strategies of performance may also be compared. Observation, experimentation, verbal protocols, and so forth are widely used in such studies. Methods for changing strategies of performance during acquisition of skills and experience are fundamental to activity theory. Individual styles of performance, derived from personal features, are also relevant (Bedny & Seglin, 1999). Error analysis, variability of time of task performance, and so forth are essential components of this approach.

### *3. ALGORITHMIC ANALYSIS OF ACTIVITY*

Algorithmic analysis of activity is a particularly powerful method of morphological analysis. It consists of the subdivision of activity into qualitatively distinct psychological units and determination of the logic of their organization and sequence. These units are formulated as elements of activity with a specific logical structure. Typically, such elements, called *members* of an algorithm, are made of actions with their associated subgoals integrated through supervening goals. Due to limits on the capacity of working memory, each member of an algorithm is limited to between one to three actions. Whereas motor actions can be performed simultaneously, mental actions are usually performed sequentially. Participants may also combine motor and cognitive actions according to the rules of temporal combination described by Bedny and Meister (1997). As units of activity, the members of algorithms are termed *operators* and *logical conditions.* Operators consist of actions that transform objects, energy, and information. For example, operators that are implicated in receiving information, analyzing a situation and comprehending it, shifting of gears, levers, and so forth can be described. Logical conditions are members of the algorithm that determine logic of selection and realization of different members of an algorithm and include a decision-making process. Human algorithms are just such algorithms defined by associated units of analysis made up of human actions. Actions as units of analysis constitute the distinctive features of a human algorithm from flow charts widely used to represent human performance.

Operative units of activity (OUA) are fundamental concepts in the study of activity. OUA is understood as contextually defined entities (image, concept, statement, comment, etc.) formed through training or experience that enable a participant to manage mentally and semantically meaningful units at levels of specificity relevant to the execution of the indicated task (Zarkovosky, Korolev, Medvedev, & Shlain, 1974). Appropriate characterization of OUA provides a great deal of leverage in developing algorithms of performance during HCI. Sometimes algorithmic description of activity may be represented as an iterative process, with sequential approaches an optimal method of performance.

Each member of the algorithm is designated by special symbols. For example, operators are represented by the symbol "*O*" and logical conditions by the symbol "*l*."

All operators involved in reception of information are categorized as afferent operators and are designated with the superscripts α, as in "$O^{\alpha}$." If the operator is involved in extracting information from long-term memory, the symbol μ is used as in $O^{\mu}$. The symbol $O^{\mu}$ is associated with keeping information in working memory, and the symbol $O^{\varepsilon}$ is associated with the executive components of activity such as the movement of a gear. Operators with the symbol $O^{\varepsilon}$ are efferent operators. In deterministic algorithms, the logical conditions designated with l have two values, zero or one. In some cases, logical conditions can be a combination of simpler ones. These simple logical conditions are connected through "and," "or," "if-then," and so forth rules. Complex logical conditions are designated by a capital "L," whereas simple logical conditions are designated by a lowercase "*l*." Logical connections between simple ones are designated with standard symbols such as "&," "^," "→," and so forth. For example, complicated logical conditions comprised from simple ones may be designated as $L_1$ ($l_1^1$ & $l_1^2$ & $l_1^3$). Symbol 1 as a subscript of capital L designates that it is the first complex logical condition. Symbol 1 as a subscript of a lowercase *l* designates that it is a simple logical condition that belongs to $L_1$. The numbers 1 through 3 are used as superscripts and designate the number of logical conditions.

In a probabilistic algorithm, logical conditions may have two or more outputs with a probability between zero and one. As a simple example, these logical conditions may be represented in the following way. Suppose the algorithm has a logical condition with three outputs with distinct probabilities of occurrence. In such a case, the logical condition can be designated as $L_1\uparrow^{1\ (1\text{–}3)}$, which possesses not two potential values but three. In this case, there are three versions of output: $\uparrow^{1\ (1)}$, $\uparrow^{1\ (2)}$, and $\uparrow^{1\ (3)}$, with different probabilities. For example, the first output has the probability .2, the second is .3, and the third is .5. Knowledge of the probability of the output may be taken into consideration in a study probability of performance of different actions, strategies of performance, calculation of the performance time of the algorithm or components of the algorithm, and evaluation of task complexity. Frequently in algorithmic description, an always-false logical condition is used, which is defined by the symbol ω. This logical condition is introduced only to make it easier to write the algorithm. It does not designate real actions performed by the participant. It always defaults to the next member of the algorithm as indicated by the arrow included in the specification of this always-false logical condition.

An arrow designates the logic of transition from one member of an algorithm to another. Thus, the algorithm exhibits all the possible actions and their logical organization and therefore constitutes a precise description of human performance. It describes activity of a participant in terms of actions through which the participant attains the goal of activity. The tabular form of the algorithm is carried out in the following way. On the left side of the table, there is a column in which the previously described symbols are placed. It is a symbolic description of the algorithm or its formula. On the right side, there is a verbal description of the members of the algorithm.

The symbols *l* or L for logical conditions in the left column include an associated arrow numbered with a superscript such as $\uparrow^2$. An arrow with the same number but a reversed position must be presented in front of another member of the algorithm to which the arrow makes reference, $\downarrow^2$. Thus, the syntax of the system is based on a semantic denotation of a system of errors and superscripted numbers. An upward pointing of a logical state of simple logical conditions *l* when $l = 1$ requires skipping the following members of the algorithm until the next appearance of the superscripted number with a downward arrow (e.g.,$\uparrow^1 \downarrow^1$). Therefore, the operator with the downward arrow with the same superscripted number in front of it is the next to be executed. If, for example, this is a probabilistic algorithm, one needs to skip the next appearance of the superscripted numbers for all possible arrows of the logical condition. Each superscripted number is associated with a discrete probability that needs to be represented as the transition process from the member of one algorithm to another.

The tabular form of an algorithm is read top to bottom. The left column with symbolic description is called the *formula* of the algorithm and is presented in a vertical orientation. In some cases, a formula for the algorithm may be presented separately as a horizontal line of symbols. In this case, the formula is read from left to right. A separate formula is used when a horizontal line is presented as an abbreviated exhibition of an algorithm. In some cases, algorithms have such large realizations that experts extract only critical ones to serve as markers for the analysis. An algorithm enables an expert to describe human performance in a probabilistic manner and uncover constraints of the work process.

Following the development of the algorithm, experts then perform psychological analysis of the algorithm, returning to a qualitative stage of analysis. Each member of an algorithm can be evaluated as a whole from qualitative and quantitative points of view. From the morphological perspective, each member of an algorithm can be described at a more detailed level in terms of actions and operations performed by humans. Actions in turn may be described in terms of the "typical elements of a task" (technological units) and "typical elements of activity" (psychological units). Typical elements of activity (psychological units) should be used during the study of HCI when researchers need a very detailed description of the structure of activity.

The relation between qualitative and algorithmic analysis of activity is not strictly linear. It is possible to transfer not only from qualitative analysis to algorithmical but also in the reverse direction. This relation between stages of analysis demonstrates principles of systemic-structural analysis of activity. When resources of qualitative stages of analysis are exhausted, the researcher then switches to algorithmic analysis. Thereafter, a qualitative analysis of a human algorithm can be performed, allowing for the correction of algorithmic description. Hence, design becomes iterative in nature.

If necessary, more detailed stages of analysis may be pursued. A designer may even proceed to a third stage of systemic-structural analysis composed of the description of the time structure of activity by using psychological units of analysis. Thereafter, the complexity of task performance can be evaluated. We do not consider these two stages in this example.

## *4. QUALITATIVE ANALYSIS OF INVENTORY RECEIVING TASK*

In this approach, the qualitative analysis starts with objectively logical analysis. It consists of a sequence of steps intimately related to algorithmic analysis. Each sequential step of the qualitative analysis is carried out in greater detail and/or for a distinct purpose. The first step is restricted to the analysis of what is currently being done, which in computer tasks is frequently quite vague or variable. The major emphasis is to identify the content of the task under investigation and its relation to other tasks. Discussion with workers or supervisors, observation, review of documents, available data compared with literature on similar work, and so forth were used at this stage. This stage enables the researcher to obtain a general understanding of technological processes and methods of work. The result of such an analysis provides a model of inventory processes for a manufacturing firm (Figure 1). The obtained data form a platform for a more specific analysis of the inventory-receiving task, henceforth to be known as *receiving*, which constitutes the major focus of this study. See from Figure 1 that the inventory process for any company consists of three subsystems: (a) stocking, (b) record keeping, and (c) WIP.

The first subsystem (Figure 1, Box 1) refers to the physical movement of items into and out of stock providing a physical quantity on hand. Raw materials, intermediate products, or finished goods are physically brought in or taken out of stock. What remains is the actual on-hand quantity—an "in" is an increase of stock, and an "out" is a decrease. An in occurs when something material is entered; an out occurs when something leaves. Stock can be increased either by purchasing or by returning items from manufacturing to stock. Stock is decreased by sale of products or component parts to customers, by putting intermediate products into manufacturing, or by scraping. When purchases are added to the stock, the stock volume increases. This is designated by a plus sign. A minus sign represents a reduction of stock.

The second subsystem (Figure 1, Box 2) represents WIP. This is a value adding manufacturing process in which diverse raw materials or intermediate products are transformed into a ready product. Movement in and out is designated the same as in the stock process.

Whenever material physically moves into or out of stock, that movement is mirrored as a transaction in the record-keeping process, which is the third component of the process model. A properly designed inventory process is capable of producing a match between the physical events that occur in Box 1 and Box 2. The record-keeping process is a complicated computerized system that has to track all physical movements of different parts, purchases, intermediate products, and so forth. The model of the inventory process depicted in Figure 1 facilitates understanding the specifics of different tasks involved in this process.

The first task is named *inventory-receiving.* Four workers responsible for registration of all purchases and movement of intermediate and final product perform this task. The task includes two parts of the job performance. One part involves physical work when a worker (later receiver) receives a box with raw material or intermediate parts. The receiver can perform two similar tasks. One task entails reception of parts from different vendors to restock the warehouse and fulfill special and emergency orders. The second task entails receiving intermediate or finished product from WIP. This study pertains to the first task.

Parts arrive at a plant in special boxes that are delivered to the reception area. Figure 2 depicts a view from above of the workplace for the receiving task. The dashed lines designate equipment introduced following the improvement of this task, which we do not discuss at this point. The number 1 represents the receiver

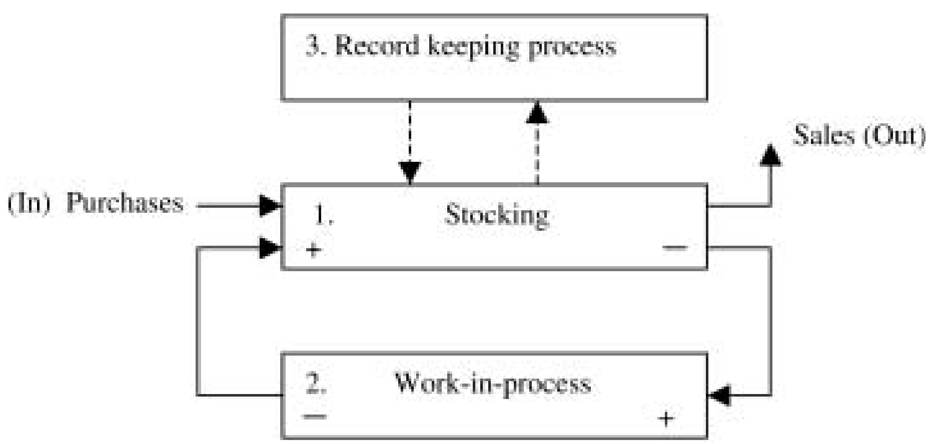


**FIGURE 1** A model of the inventory process for a manufacturing firm before improvement. 1 = Route 1 goes from (in) purchase to stocking then to sales (out) or WIP.

who opens the boxes placed on the base unit (5). For this purpose, the receiver uses a special knife. After opening the box, the receiver removes a packing slip and reads it. Then the receiver uses a computer-based warehouse management system (3). The receiver enters the purchase order (PO) number listed on the packing slip and hits the F3 key to check what is still open on the PO. The receiver takes the parts out of the box and compares the order quantity with received quantity. The receiver chooses the item from the PO then changes or confirms the quantity and the price and assigns allocation if necessary. If allocation is already reserved for the item, the system will select it automatically. All required information is shown on the screen; later this information is printed on the label.

One can specify two kinds of subtasks; the first is the setup subtask, and the second is the main subtask. The setup operation includes login, menu selection, key in PO number, and so forth. The main operation begins when an item is taken out of the box and ends when it is put in the tote. The receiver places each part from the vendor into a tote. The tote thus filled with parts is placed in a put-away area by the stock Belt 8. The second task is putting away. The put-away operator takes parts from the tote and places them on the corresponding shelves.

The next task is pickup. The pickup operator takes the parts that have been ordered from the shelves and places them in the tote. This tote is later delivered to the

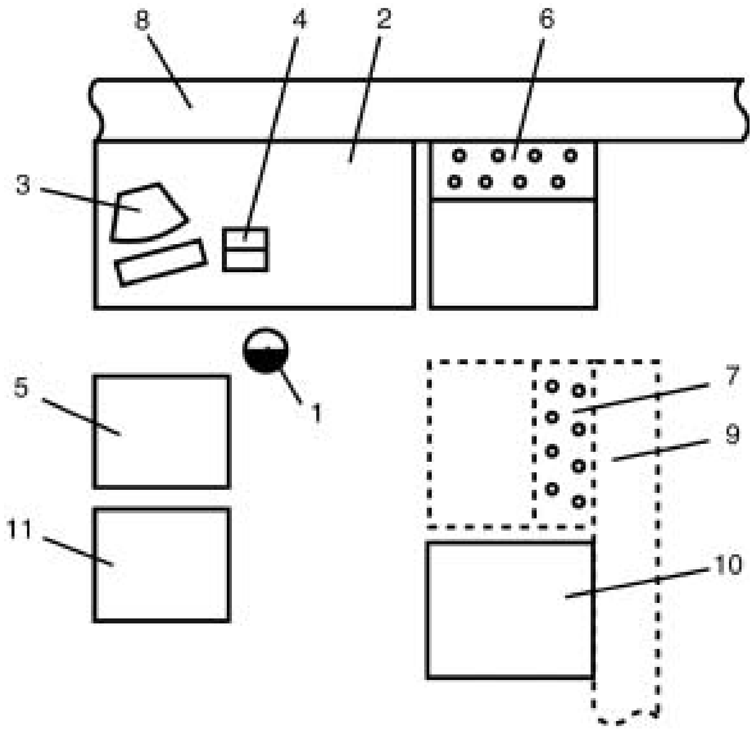


**FIGURE 2** A view of the receiving task from above the workplace. 1 = receiver; 2 = work table; 3 = computer; 4 = tag printer; 5 = base unit for unpacking; 6 = base unit for stock process; 7 = base unit for work-in-process (WIP); 8 = belt for stocking; 9 = belt for WIP; 10 = put aside area; 11 = place for tote.

workshop for production. The pickup operator also places ready-for-sale products into the tote. Delivering of the required parts for production is related to the delivery task. Figure 3 designates the sequence of operations just described before the improvement. These operations are covered in relation to the receiving task.

In the previous section, we give a brief description of the receiving task and those tasks that attend it. The qualitative analysis may be labeled as analysis with a technological orientation. The data gathered by qualitative-technological analysis (the first step of qualitative analysis) may now be used for the second stage of analysis called *algorithmic analysis.*

## 5. ALGORITHMIC DESCRIPTION OF THE RECEIVING TASK BEFORE ITS IMPROVEMENT

The data discussed in the previous section merely provides preliminary data related to qualitative analysis. To perform more in-depth qualitative analysis, more detailed information is needed. Accordingly, a second stage of analysis using algorithmic description is called for. Only after this second stage may a qualitative analysis be revisited with greater elaboration and insight. The algorithmic model of activity during the performance of the receiving task is presented in Table 1 (only those actions intimately related to computer use are described). In Table 1, the left column delineates the symbolic model of the activity. The right column of Table 1 contains a verbal description of the algorithm. A short description of the development of the algorithm was offered in the preceding preliminary sections. Table 1 reveals that for the description of human activity, a probabilistic algorithm was used rather than a deterministic one. Actually, many logical conditions possess more than two outputs. Moreover, each output can possess different probabilities. Hence, this task has different degrees of uncertainty. Progressive reading of this algorithm from top to bottom and comparison of each member of the algorithm with information presented on the computer screen enables one to see a precise picture of how users carried out the computer-mediated task.

Reading each member of an algorithm in symbolic representation allows one to understand the logic of the transition from one member of an algorithm to another. Algorithmic description also provides insight into the psychological peculiarities of each member of the algorithm. For example, $O^{\alpha}$ indicates that this member of the algorithm refers to perceptual actions, $l^{\mu}_{15}\uparrow^{(1-10)}$ exhibits decision-making actions with 10 outputs and shows its involvement in intensive utilization of memory and so forth. Figures 4 and 5 are examples of screens used in the algorithmic description of activity.

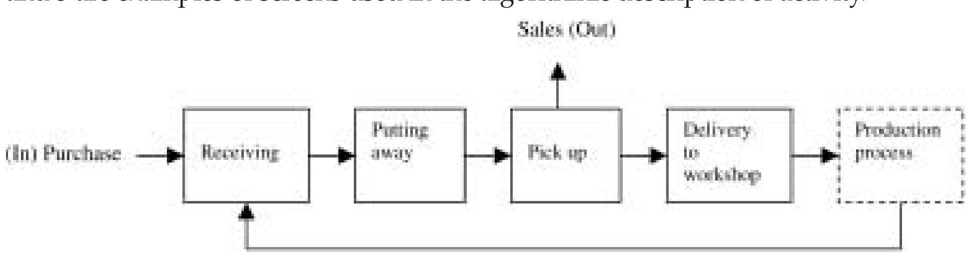


**FIGURE 3** The sequence of tasks before the improvement.

Algorithmic description of activity entails much effort and time. Nevertheless, note that many people repeat some tasks over periods of decades, which in the absence of a precise formulation of human performance results in user unfriendly software designs. Currently, improvement and innovation are gates to the user who, lacking a paradigm for conceptualizing the process, is plunged into endless iterative cycles of trial and error. The precise algorithmic description of the task in combination with qualitative analysis can significantly reduce cycle time. The algorithmic description we presented allows revisiting of the preliminary stages of analysis.

## 6. SECOND STEP OF THE QUALITATIVE ANALYSIS OF AN INVENTORY RECEIVING TASK

At this step, the receiving task was formulated at a more detailed level. Whereas the first step focused on present circumstances, the second step emphasized locating deficiencies in the existing task, identifying psychological difficulties of performance, quality analysis, and so forth. It was discovered that tasks are multivariant and cannot be treated as deterministic. Subjective difficulties of performance were noted, such as specificity of interactions with others and supervisors. This means that the sociocultural context of work assumes importance. The objective of the study was not merely the task as a whole. Rather, each member of the algorithm was studied as a quasi-system. The logic of the transition from one member of an algorithm to another was also defined. Comparing this data with direct observation facilitates the discovery of many deficiencies in the task and in inventory process in general. The model of the inventory receiving process (Figure 1) and algorithm of performance (Table 1) suggested that all parts received from vendors should be directly placed in storage on warehouse shelves from which they can be delivered on request to the various workshops. However, there were instances in which observation or deliberations called for immediate deployment of items into the manufacturing process. These parts should be treated as special order items. A model of an inventory process that encompasses this latter contingency is illustrated in Figure 6. This model shows that received parts have two possible routes: one proceeds to stock, whereas the other routes to WIP.

Work procedures prior to improvement were performed according to the model of inventory process that is exhibited in Figure 1. According to this model, the task unfolds as in Figure 3. The task "delivery to workshop" may be carried out only after the receiving, putting away, and pickup tasks are performed. This results in a delay of production process, unnecessary work, and performance under rushed, stressful circumstances. It has been empirically discovered that more than 20% of parts should be sent immediately into workshops. Thus, the model of inventory process depicted in Figure 1 fails to correspond to the real requirements of a production process. It ignores the situations in which there is an emergency order and parts from vendors have to be delivered immediately to production, that is, WIP. Therefore, the second model on Figure 6 was recommended. With this process, roughly 20% of the parts go immediately to production, whereas the majority of parts route to stock. Under this model, a different sequence of operations is depicted (Figure 7). Figure 7 shows that the receiver is required to place some parts directly into the delivery to workshop

**Table 1: Algorithmic Description of Activity During Computer Based Task Performance (Before Improvement)**

| *Member of Algorithm* | *Description of Members of Human Algorithm* |
|---|---|
| $O^{\alpha}_{1}$ | Check for presence of inventory receiving screen |
| $\overset{1}{\downarrow}O^{\varepsilon}_{2}$ | Type 1 and then press ENTER to choose ADD INVENTORY RECEIVING screen (see Figure 4) |
| $O^{\alpha}_{3}$ | Check to see if you are at the ADD TRANSACTION screen (cursor on Field 1) |
| $l_1\overset{1}{\uparrow}$ | If you are at the right screen, go to $O^{\varepsilon}_{4}$; if the screen is wrong, hit F3 for exit and go back to $O^{\varepsilon}_{2}$ |
| $\overset{17(1)}{\downarrow}\overset{9(1)}{\downarrow}\overset{7(1)}{\downarrow}O^{\varepsilon}_{4}$ | Take a packing slip from the box placed on Base Unit 5 (see Figure 2) |
| $O^{\alpha}_{5}$ | Find purchase order (PO) number on the slip |
| $O^{\varepsilon}_{6}$ | Key in PO number and hit enter (Figure 6, Field 1) |
| $O^{\alpha}_{7}$ | Look at the screen message |
| $l_2\overset{2}{\uparrow}$ | If the screen displays an error message INVALID PO NUMBER, then go to operator $O^{\alpha}_{8}$; if PO number is correct, the cursor moves to the second field RECV-DATE (see Figure 4), go to $O^{\varepsilon}_{11}$ |
| $O^{\alpha}_{8}$ | Compare PO number on the screen with the number on the packing slip |
| $l_3\overset{3}{\uparrow}$ | If PO number does not match, go to $O^{\varepsilon}_{9}$; if the PO number is correct and error message persists (system can not find purchase order) go to $O^{\varepsilon}_{10}$ |
| $O^{\varepsilon}_{9}$ | Key in the correct number again |
| $\overset{3}{\downarrow}O^{\varepsilon}_{10}$ | Call manager |
| $\overset{2}{\downarrow}O^{\varepsilon}_{11}$ | Key in a current date or the date it has been received (the cursor moves to Field 2; see Figure 4) |
| $\overset{17(2)}{\downarrow}\overset{11(2)}{\downarrow}\overset{7(2)}{\downarrow}O^{\varepsilon}_{12}$ | Press F8 to look up items on the PO (Figure 4) |
| $O^{\varepsilon}_{13}$ | Take out item from Box 5 |
| $O^{\alpha}_{14}$ * | Look at item number and compare it with item numbers (Figure 4, Field 3) on the screen |
| $l_4\overset{4}{\uparrow}$ | If item number is on the first page, go to $O^{\varepsilon}_{16}$; if item number is not on the first page, go to $O^{\varepsilon}_{15}$ |
| $O^{\varepsilon}_{15}$ | Hit arrow key (repeat if required) |
| $\overset{4}{\downarrow}O^{\varepsilon}_{16}$ | Put cursor on the selected line (Figure 4) and hit ENTER to go to the screen with detail item information (Figure 5) |
| $\overset{4}{\downarrow}O^{\alpha}_{17}$ | Compare received quantity with PO quantity (Figure 5, Field 4) |
| $\overset{\alpha th}{l}_{5}\overset{5}{\uparrow}$ ** | If received quantity and ordered quantity are the same, press ENTER and go to $O^{\varepsilon}_{24}$; if received quantity is greater or less than ordered quantity, go to $O^{\varepsilon}_{19}$ |
| $O^{\varepsilon}_{19}$ | Type the received quantity and press ENTER to get a question on the bottom of the screen |
| $O^{\alpha}_{20}$ | Read the statement: THE RECIVED QUANTITY AND ORDERED QUANTITY DOES NOT MATCH. DO YOU ACCEPT? (YES/NO) |

*(continued)*

**Table 1 (Continued)**

| *Member of Algorithm* | *Description of Members of Human Algorithm* |
|---|---|
| $l^{th}_{6}\overset{6}{\uparrow}$ | If quantity is not accepted (computer defaults to "N") go to $O^{\varepsilon}_{21}$; otherwise, go to $O^{\varepsilon}_{23}$ |
| $O^{\varepsilon}_{21}$ | Press ENTER |
| $O^{\alpha}_{22}$ | Check if there are other items on this PO to receive |
| $l_{7}\overset{7(1-2)}{\uparrow}$ | If there are no more items in the box, go to $O^{\varepsilon}_{4}$, otherwise go to $O^{\varepsilon}_{12}$ |
| $\overset{6}{\downarrow}O^{\varepsilon}_{23}$ | Type "Y", press ENTER |
| $\overset{5}{\downarrow}O^{\varepsilon}_{24}$ | Compare price of the item on the shipping list with price on the screen |
| $l_{8}\overset{8}{\uparrow}$ | If the price on the screen and shipping list are different, go to $O^{\varepsilon}_{25}$; otherwise, go to ${}^{2}O^{\varepsilon}_{31}$ |
| $O^{\varepsilon}_{25}$ | Key in the new price and hit ENTER |
| $O^{\alpha}_{26}$ | Read the message, THE PRICE YOU ENTERED DOES NOT MATCH INITIAL PRICE. DO YOU WANT TO ACCEPT? (Y/N) on the screen (Figure 5, Field 5) |
| $O_{27}^{\alpha\,th}$ *** | Compare new price with ordered price |
| $l_{9}\overset{9}{\uparrow}$ | If new price is smaller or equal, go to ${}^{1}O^{\varepsilon}_{31}$; if new price is greater go to $O_{28}^{th\,\mu}$ |
| $O_{28}^{th\,\mu}$ | Mentally calculate the price difference |
| $l^{\mu}_{10}\overset{10}{\uparrow}$ | If difference is less than 10%, go to ${}^{1}O^{\varepsilon}_{31}$; if difference is greater than 10%, go to ${}^{1}O^{\varepsilon}_{29}$ (unless instructed otherwise) |
| $O^{\varepsilon}_{29}$ | Type "N" and hit ENTER (the item is put aside and task is completed) |
| $O^{\alpha}_{30}$ | Check if there are other items in this box to receive |
| $l_{11}\overset{11(1-2)}{\uparrow}$ | If there are no more items to receive, go to $O^{\varepsilon}_{4}$; otherwise go to $O^{\varepsilon}_{12}$ |
| $\overset{10}{\downarrow}\ \overset{9}{\downarrow}\ {}^{1}O^{\varepsilon}_{31}$ | Type "Y" |
| $\overset{8}{\downarrow}\ {}^{2}O^{\varepsilon}_{31}$ | Hit ENTER to go to the Completion Flag field (Figure 5, Field 6) |
| $O^{\alpha}_{32}$ | Check system default (Y/N) "flag;" system gives default according to the rule "If received quantity $\geq$ ordered quantity system defaults to 'Y' otherwise it defaults to 'N'" |
| $l_{12}\overset{12}{\uparrow}$ | If you except the system default (Y/N), go to $O^{\varepsilon}_{34}$; otherwise go to $O^{\varepsilon}_{33}$ |
| $O^{\varepsilon}_{33}$ | If system defaults to "N", type "Y", and go to $O^{\varepsilon}_{34}$; if system defaults to "Y", type "N", and go to $O^{\varepsilon}_{34}$ |
| $\overset{12}{\downarrow}O^{\varepsilon}_{34}$ | Hit ENTER to go to the next field (Figure 5, Field 7) |
| $O^{\alpha}_{35}$ | Check if there is a bin for this item |
| $l_{13}\overset{13}{\uparrow}$ | If the bin is not assigned for this item, go to $O^{\mu}_{36}$; otherwise the system will automatically assign the required bin then go to $O^{\varepsilon}_{40}$ |
| $\overset{15(1)}{\downarrow}O^{\mu}_{36}$ | Depending on the size, shape and special features ("HazMat") of item, recall required bin type |

*(continued)*

**Table 1 (Continued)**

| *Member of Algorithm* | *Description of Members of Human Algorithm* |
|---|---|
| $l^{\mu}_{14}\ \overset{14(1-10)}{\downarrow}$ | If bin type is 1, go to $^{1}O^{\varepsilon}_{37}$, up to "If bin type is 10," go to $^{10}O^{\varepsilon}_{37}$ |
| $\overset{16(1)}{\downarrow}\ \overset{14(1)}{\downarrow}\ {}^{1}O^{\varepsilon}_{37}$ | Type "1" and hit ENTER |
| $\omega_1 \uparrow_{\omega 1}$ **** | Always falls logical conditions (see $O^{\alpha}_{38}$) |
| . . . . | Choose the required bin type |
| $\overset{16(10)}{\downarrow}\ \overset{14(10)}{\downarrow}\ {}^{1}O^{\varepsilon}_{37}$ | Type "10" and hit ENTER |
| $\downarrow_{\omega(1-9)} O^{\alpha}_{38}$ | Check error message on the screen ($O^{\alpha}_{38}$ follow after every $O^{\varepsilon}_{37}$) |
| $l_{15}\ \overset{15(1-2)}{\uparrow}$ | If you get an error message, THIS IS A WRONG BIN TYPE, go to $O^{\alpha}_{39}$ or $O^{\mu}_{36}$; otherwise go to $O^{\varepsilon}_{40}$ |
| $O^{\alpha}_{39}$ | Look at the bin type chart |
| $l_{16}\overset{16}{\uparrow}$ | If bin type is 1, go to $^{1}O^{\varepsilon}_{37}$; if bin type is 10, go to $^{10}O^{\varepsilon}_{37}$ |
| $\overset{15(2)}{\downarrow}\ \overset{3}{\downarrow} O^{\varepsilon}_{40}$ | Hit ENTER to print the label |
| $O^{\varepsilon}_{41}$ | Peel the label off the printer and put it on the part |
| $O^{\varepsilon}_{42}$ | Put part in the tote |
| $O^{\alpha}_{43}$ | Check if there are other items in the box to receive |
| $l_{17}\ \overset{17(1-2)}{\uparrow}$ | If there are no more items to receive, go to $O^{\varepsilon}_{4}$, otherwise go to $O^{\varepsilon}_{12}$; if there are no new boxes to work with, go to $O^{\varepsilon}_{44}$ |
| $O^{\varepsilon}_{44}$ | Hit F3 and go to the previous screen |
| $\overset{18}{\downarrow}O^{\varepsilon}_{45}$ | Type "3" and then press ENTER to choose PRINT REPORT |
| $O^{\alpha}_{46}$ | Check to see if you are at the PRINT REPORT screen |
| $l_{18}\overset{18}{\uparrow}$ | If you are at the right screen, go to $O^{\varepsilon}_{47;}$ if the screen is wrong, hit F3 for EXIT and go back to $O^{\varepsilon}_{45}$ |
| $O^{\varepsilon}_{47}$ | Key in Start Date, hit ENTER, key in End Date, hit ENTER |
| $O^{\varepsilon}_{48}$ | Wait for the report completion message |

*One PO usually required no more than 3 passes on the screen. **$l^{\alpha th}$ stands for "thinking actions" involved in decision making that are performed based on visual information. ***$O_{27}{}^{\alpha th}$ stands for executive "thinking actions" that are performed based on visual information. ****ω always falls logical condi-

tote, whereas the majority of the parts should be directed to stock bins. Hence, the content of the receiving task should be changed. In addition, the receiver has to determine what category of the order he or she is processing by answering the question "WORK-IN-PROCESS (Y/N)?" on the screen (see Figure 8).

The algorithm of performance facilitated determination of which step to introduce decision rules regarding assignment of parts and new actions related to the WIP. The analysis of the logic of the performance algorithm revealed that these actions should be introduced following checking the purchase order number and

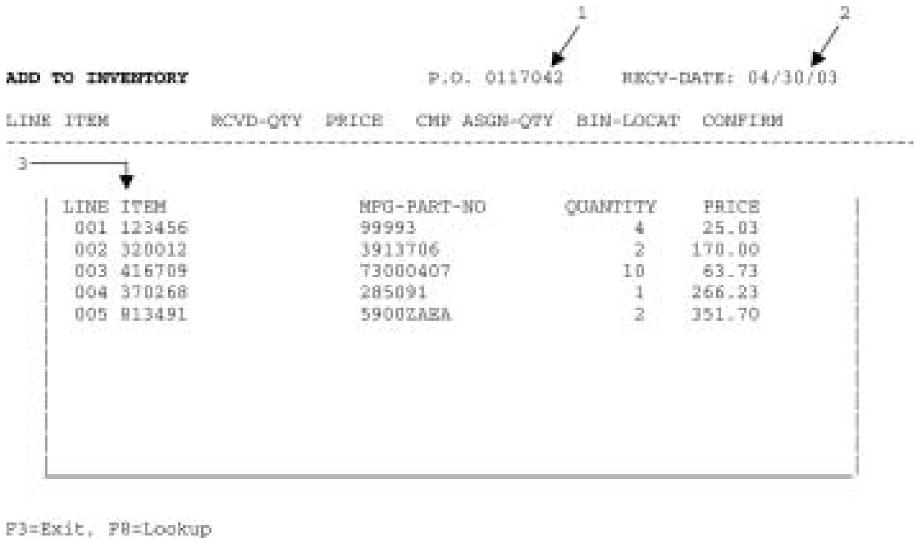


**FIGURE 4** Add inventory receiving screen. 1 = purchase order number; 2 = received date; 3 = item number.

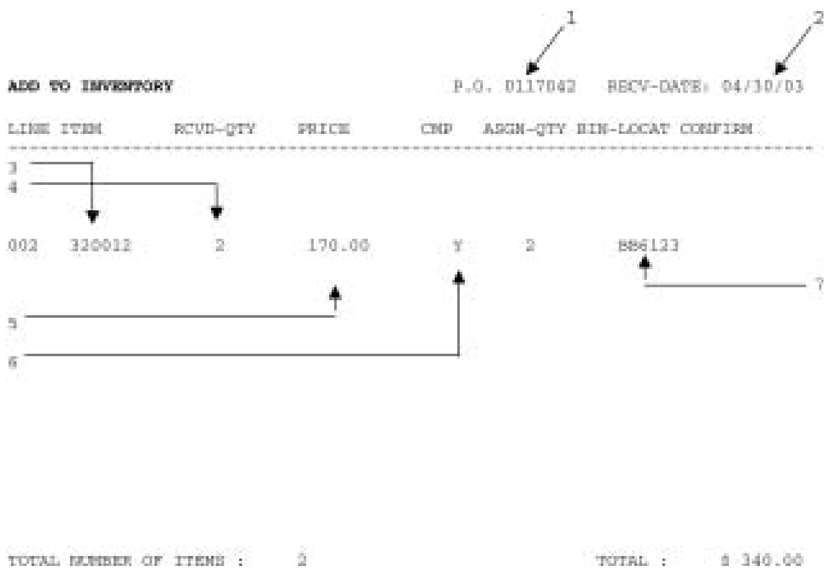


**FIGURE 5** Add receiving screen with detailed item information. 1 = purchase order number; 2 = received date; 3 = item number; 4 = received quantity; 5 = unit cost; 6 = competition flag; 7 = bin location.

item number and comparing the received quantity and price. In terms of the algorithm, this meant that introduction of decision rules should be inserted after $O^{\varepsilon}_{34}$ and before $O^{\alpha}_{35}$ in Table 1. In Table 2 (algorithmic description of task after improvement), members of the algorithm from $O^{\alpha}_{33}$ to $O^{\varepsilon}_{36}$ describe components of activity related to WIP.

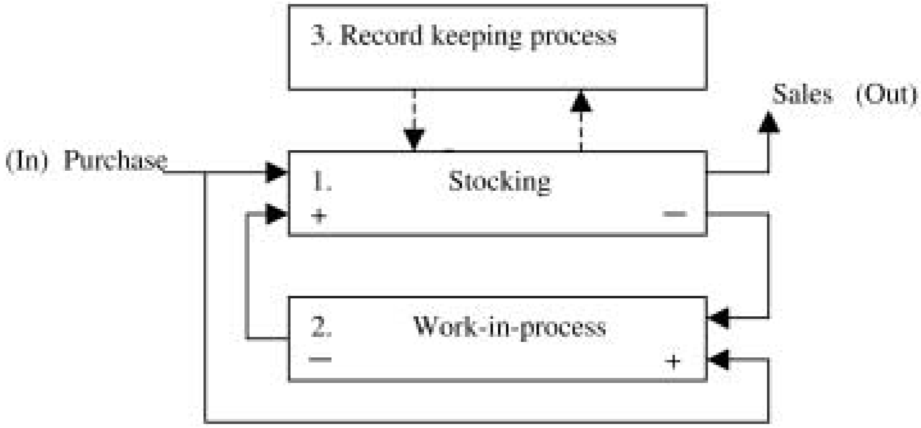


**FIGURE 6** A model of the inventory process for a manufacturing firm after improvement. 1 = Route 1 goes from (in) purchase to stocking then to sales (out) or work-in-process (WIP); 2 = Route 2 goes from (in) purchase to WIP then to stocking and sales (out) or again to WIP.

The method for developing an algorithm following improvement in this case was the same as described previously. The difference here was that the researcher had the opportunity at this stage to compare preliminarily performed qualitative analysis and algorithmic description of the task in Table 1. Accordingly, comparative analysis follows immediately, bypassing development of a second version of the algorithm presented in Table 2.

## 7. COMPARATIVE ANALYSIS OF ACTIVITY ALGORITHMS BEFORE AND AFTER IMPROVEMENT OF THE RECEIVING TASK

The comparative analysis began with a study of WIP improvement. Consider Table 2 in which WIP is presented algorithmically. Members of the algorithm from $O^{\alpha}_{33}$ through $O^{\varepsilon}_{36}$ define the content and logic of actions performed by a receiver if the screen displays "work-in-process" and the answer is "Yes" (Y). If, according to the received message, the answer is a "No" (N), the receiver bypasses all previously mentioned members of the algorithm and performs the tasks in a regular order. It is clear that if computer programmers re-

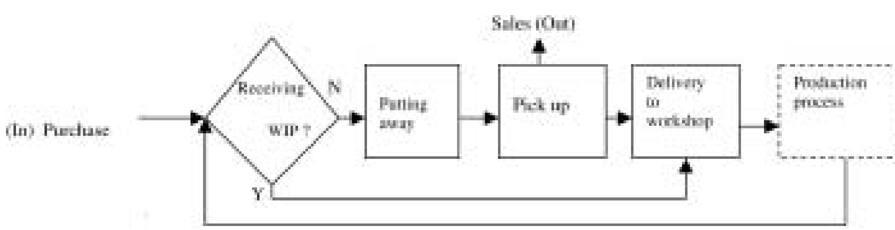


**FIGURE 7** The sequence of tasks after the improvement. 1 = Route 1 receiving and question work-in-process (WIP; response "N") then putting away up to production process; 2 = Route 2 receiving and question WIP (response "Y") then delivery to workshop up to production process.

ceived a precise description of the actions performed by users related to WIP they could introduce more efficient changes in the design of the software, providing for a new way of task performance. After obtaining a clear and precise description of the actions performed by a user, a programmer could develop programs that require a minimum of corrections and debugging. It is, of course, well known that because users frequently are unable to explicate the task requirements to programmers, software design expands into a long sequence of improvements. Moreover, users often change their opinion after improvements. Thus, algorithmic description of human activity during task performance enables evaluation of the efficiency of user actions. Actions performed by users according to the algorithmic representation become clearly understandable if the algorithmic description of the task and workplace arrangements are compared. According to the algorithmic description in Table 2 and the work arrangement in Figure 2, if a receiver gives an answer on the screen "work-in-process"—"No," he or she uses Base Unit 6 (in Figure 2) for the stock process and Belt 8. If a user's answer for "work-in-process" is "Yes," he or she uses Base Unit 7 for WIP and Belt 9. Base Unit 7 and Belt 9 were introduced after improvement as designated by the dashed lines in Figure 2.

Thus, the fragment of the algorithm introduced as improvement adds to "up-front" work but eliminates subsequent steps of the algorithm altogether, thereby reducing the overall task burden. It is also noteworthy that in the 20% of cases that route directly to WIP, the unnecessary tasks (putting away, pickup) are completely eliminated. The following stages of analysis involving evaluation of task complexity before and after improvement indicate that there is a negligible increase in task complexity for some subtasks. This is attributable to additional afferent operator $O^{\alpha}_{33}$ and logical condition $l_{12}$. Finally, some measures of variability of task performance may be slightly increased, which is offset by the elimination of unnecessary work and reduction of time constraints.

We consider the following steps involved in comparative analysis of different members of the algorithms before and after improvement. For discussion purposes only, the more important members of the algorithm that are germane to performance improvement against the baseline are selected. We compare a member of algorithm $O^{\varepsilon}_{11}$ before and after improvement. Before improvement, it is revealed that the worker must recall the current date and key it in. After the improvement, the worker must simply hit enter and the system defaults to the current date. Logical condition $l^{th}_{6}$ includes reasoning ac-

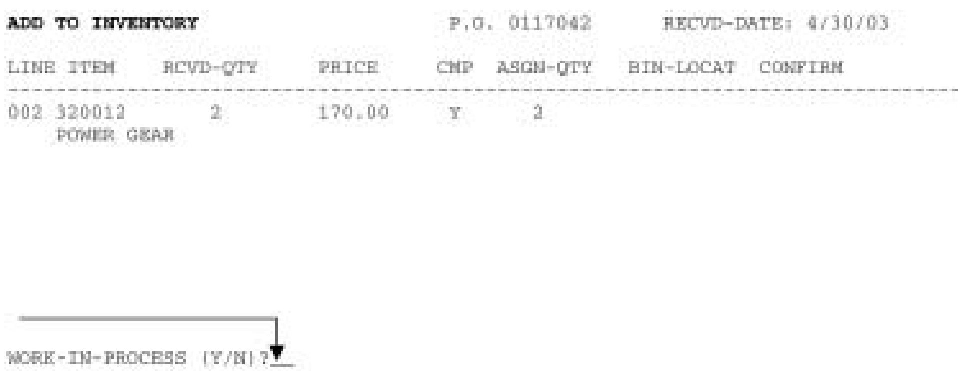


**FIGURE 8** Screen with the work-in-process option. 1 = yes or no answer is required.

**Table 2: Algorithmic Description of Activity During Computer Based Task Performance (After Improvement).**

| *Member of Algorithm* | *Description of Members of Human Algorithm* |
|---|---|
| $O^{\alpha}_{1}$ | Check for presence of inventory receiving screen |
| $\overset{1}{\downarrow}O^{\varepsilon}_{2}$ | Type 1 and then press ENTER to choose ADD INVENTORY RECEIVING screen |
| $O^{\alpha}_{3}$ | Check to see if you are at the ADD INVENTORY RECEIVING screen (cursor in Field 1) |
| $l_{1}\overset{1}{\uparrow}$ | If you are at the right screen, go to operator $O^{\varepsilon}_{4}$; if the screen is wrong, hit F3 for EXIT and go back to $O^{\varepsilon}_{2}$. |
| $\overset{16(1)}{\downarrow}\overset{10(1)}{\downarrow}\overset{7(1)}{\downarrow}O^{\varepsilon}_{4}$ | Take a packing slip from the box placed on the Base Unit 5 (see Figure 2) |
| $O^{\alpha}_{5}$ | Find purchase order (PO) number on the slip |
| $O^{\varepsilon}_{6}$ | Key in PO number and hit enter (Figure 4, Field 1) |
| $O^{\alpha}_{7}$ | Look at the screen message |
| $l_{2}\overset{2}{\uparrow}$ | If the screen displays an error message, INVALID PO NUMBER, then go to operator $O^{\alpha}_{8}$; if PO number is correct, (the cursor moves to the second field RECV-DATE, see Figure 4) go to $O^{\varepsilon}_{11}$ |
| $O^{\alpha}_{8}$ | Compare PO number on the screen with the number on the packing slip |
| $l_{3}\overset{3}{\uparrow}$ | If PO number does not match go to $O^{\varepsilon}_{9}$. If the PO number is correct and error message persists (system can not find purchase order), go to $O^{\varepsilon}_{10}$ |
| $O^{\varepsilon}_{9}$ | Key in the correct number again |
| $\overset{3}{\downarrow}O^{\varepsilon}_{10}$ | Call manager |
| $\overset{2}{\downarrow}O^{\varepsilon}_{11}$ | Hit ENTER to get a current date (the cursor moves to Field 2; see Figure 4) |
| $\overset{16(2)}{\downarrow}\overset{10(2)}{\downarrow}\overset{7(2)}{\downarrow}O^{\varepsilon}_{12}$ | Press F8 to look up items on the purchase order (Figure 4, Field 3) |
| $O^{\varepsilon}_{13}$ | Take out item from Box 5 |
| $O^{\alpha}_{14}$ | Look at item number and compare it with item number (Figure 4, Field 3) on page 1 on the screen |
| $l_{4}\overset{4}{\uparrow}$ | If item number is on the first page, go to $O^{\varepsilon}_{16}$;if item number is not on the first page, go to $O^{\varepsilon}_{15}$ |
| $O^{\varepsilon}_{15}$ | Hit arrow key (repeat if required) |
| $\overset{4}{\downarrow}O^{\varepsilon}_{16}$ | Put cursor on the selected line (Figure 4) and hit ENTER to go to the screen with detail information (Figure 5) |
| $\overset{4}{\downarrow}O^{\alpha}_{17}$ | Compare received quantity with PO quantity (Figure 5, Field 4) |
| $l^{\alpha\,th}_{5}\overset{5}{\uparrow}$ | If received quantity and ordered quantity are the same, press ENTER and go to $O^{\varepsilon}_{24}$; if received quantity is greater or less than ordered quantity, go to $O^{\varepsilon}_{19}$ |
| $O^{\varepsilon}_{19}$ | Type the received quantity and press enter to get a question on the bottom of the screen |
| $O^{\alpha\,th}_{20}$ | Read the statement, THE RECEIVED QUANTITY DOES NOT MATCH ORDERED QUANTITY. Do you want to accept? (Yes/No) |

*(continued)*

**Table 2 (Continued)**

| *Member of Algorithm* | *Description of Members of Human Algorithm* |
|---|---|
| $l_{6\uparrow}^{6}$ | If quantity is not accepted (computer defaults to N) go to $O^{\varepsilon}_{21}$; otherwise, go to $O^{\varepsilon}_{23}$ |
| $O^{\varepsilon}_{21}$ | Press ENTER |
| $O^{\alpha}_{22}$ | Check if there are other items in the box to receive |
| $l_{7\uparrow}^{7(1-2)}$ | If no more items in the box, go to $O^{\varepsilon}_{4}$; otherwise go to $O^{\varepsilon}_{12}$ |
| ${}^{6}_{\downarrow}O^{\varepsilon}_{23}$ | Type "Y," press ENTER |
| ${}^{5}_{\downarrow}O^{\alpha}_{24}$ | Compare price of the item on the shipping list with price on the screen |
| $l_{8\uparrow}^{8}$ | If the price on the screen and shipping list are different, go to $O^{\varepsilon}_{25}$; otherwise, go to ${}^{2}O^{\varepsilon}_{29}$ |
| $O^{\varepsilon}_{25}$ | Key in the new price and hit ENTER |
| $O^{\alpha}_{26}$ | Look at information on the screen; cursor can move to the next field, or the message, PRICE DIFFERENCE IS GREATER THAN 10%. DO YOU WISH TO PROCEED? (Y/N) can be presented |
| $l_{9\uparrow}^{9(1-2)}$ | If the message, "PRICE DIFFERENCE IS GREATER THAN 10%. DO YOU WISH TO PROCEED? (Y/N)" appears and the answer is "N," go to $O^{\varepsilon}_{27}$; if "Y" (special instruction) go to ${}^{1}O^{\varepsilon}_{29}$; if there is no message (cursor moved to the next field), go to $O^{\alpha}_{30}$ |
| $O^{\varepsilon}_{27}$ | Type "N" and hit ENTER |
| $\underset{\omega 1}{\downarrow}\underset{\omega 2}{\downarrow}\underset{\omega 3}{\downarrow}O^{\alpha}_{28}$ | Check if there are other items in this box to receive |
| $l_{10\uparrow}^{10(1-2)}$ | If no more items to receive, go to $O^{\varepsilon}_{4}$; otherwise, go to $O^{\varepsilon}_{12}$ |
| ${}^{9(1)}_{\downarrow}\,{}^{1}O^{\varepsilon}_{29}$ | Type "Y" |
| ${}^{8}_{\downarrow}\,{}^{2}O^{\varepsilon}_{29}$ | Hit ENTER to go to the Completion Flag field (Figure 5, Field 6) |
| ${}^{9(2)}_{\downarrow}O^{\alpha}_{30}$ | Check system default (Y/N ) (system gives a default according to the rule, "If received quantity ≥ ordered quantity, system defaults to Y;' otherwise it defaults to 'N'" |
| $l_{11\uparrow}^{11}$ | If you except system default (Y/N), go to $O^{\varepsilon}_{32}$, otherwise, go to $O^{\varepsilon}_{31}$ |
| $O^{\varepsilon}_{31}$ | System defaults to "N," type "Y," and go to $O^{\varepsilon}_{32}$. System defaulted to "Y," type "N" and go to $O^{\varepsilon}_{32}$ |
| ${}^{11}_{\downarrow}O^{\varepsilon}_{32}$ | Hit ENTER to go to the next field (Figure 5, Field 7) |
| $O^{\alpha}_{33}$ | Look at the screen message |
| $l_{12\uparrow}^{12}$ | If screen displays a message, WORK-IN-PROCESS? (Y/N) and the answer is "Y" go to $O^{\varepsilon}_{34}$; otherwise, go to $O^{\varepsilon}_{37}$ (Figure 8) |
| $O^{\varepsilon}_{34}$ | Type 'Y" hit ENTER to print out a label, and put label on the part |
| $O^{\alpha}_{35}$ | Look at the label to determine which department within the plant the item will be shipped to |
| $l_{13\uparrow}^{13(1-3)}$ | If it goes to Department 1, go to ${}_{1}O^{\varepsilon}_{36}$ and if it goes to Department 2, go to ${}_{2}O^{\varepsilon}_{36}$; otherwise, go to ${}_{3}O^{\varepsilon}_{36}$ |

*(continued)*

**Table 2 (Continued)**

| *Member of Algorithm* | *Description of Members of Human Algorithm* |
|---|---|
| $\overset{13(1)}{\downarrow}\ {}_{1}O^{\varepsilon}_{36}$ | Put the part in Box 1 |
| $\omega_{1}\uparrow_{\omega 1}$ | Always falls logical condition (see $O^{\alpha}_{28}$) |
| $\overset{13(2)}{\downarrow}\ {}_{2}O^{\varepsilon}_{36}$ | Put the part in Box 2 |
| $\omega_{2}\uparrow_{\omega 2}$ | Always falls logical condition (see $O^{\alpha}_{28}$) |
| $\overset{13(3)}{\downarrow}\ {}_{2}O^{\varepsilon}_{36}$ | Put the part in Box 3 |
| $\omega_{3}\uparrow_{\omega 3}$ | Always falls logical condition (see $O^{\alpha}_{28}$) |
| $\overset{12}{\downarrow}O^{\alpha}_{37}$ | Check if there is a bin for this item |
| $l_{14}\overset{14}{\uparrow}$ | If the bin is not assigned for this item, go to $O^{\varepsilon}_{38}$; otherwise the system will automatically assign the required bin, then go to $O^{\varepsilon}_{41}$ |
| $O^{\alpha}_{38}$ | Depending on the size, shape, and special features ("HazMat"), choose the bin type from the list of bin types on the screen |
| $l_{15}\overset{15(1...10)}{\uparrow}$ | If bin is Type 1, go to ${}^{1}O^{\varepsilon}_{39}$ up to "If bin type 10," go to ${}^{10}O^{\varepsilon}_{39}$ |
| $\overset{15(1)}{\downarrow}\ {}^{1}O^{\varepsilon}_{39}$ | Choose the bin Type 1 (move cursor to required position and hit ENTER) |
| $\omega_{4}\uparrow_{\omega 4}$ | Always falls logical condition |
| •<br>•<br>• | Choose the required bin type |
| $\overset{15(10)}{\downarrow}\ {}^{10}O^{\varepsilon}_{39}$ | Choose the bin Type 10 |
| $\downarrow_{\omega 4}O^{\varepsilon}_{40}$ | Hit ENTER (system will assign the available bin of the chosen type) |
| $\overset{14}{\downarrow}O^{\varepsilon}_{41}$ | Hit ENTER to print the label |
| $O^{\varepsilon}_{42}$ | Peel the label off the printer and put it on the part |
| $O^{\varepsilon}_{43}$ | Put part in the tote |
| $O^{\alpha}_{44}$ | Check if there are other items in the box to receive |
| $l_{16}\overset{16(1-2)}{\uparrow}$ | If there are no more items to receive, go to $O^{\varepsilon}_{4}$, otherwise go to $O^{\varepsilon}_{12}$; if there are no new boxes to work with, go to $O^{\varepsilon}_{45}$ |
| $\overset{17(2)}{\downarrow}O^{\varepsilon}_{45}$ | Hit F3 and go to the previous screen |
| $O^{\alpha}_{46}$ | Check for presence of inventory receiving screen |
| $\overset{17}{\downarrow}O^{\varepsilon}_{47}$ | Type 3 and then press ENTER to choose PRINT REPORT |
| $O^{\alpha}_{48}$ | Check to see if you are at the PRINT RESEIVING REPORT screen |
| $l_{17}\overset{17(1-2)}{\uparrow}$ | If you are at the right screen and you choose current date, go to $O^{\varepsilon}_{49}$; if you choose a different date range (From–To) go to $O^{\varepsilon}_{50}$; if you are at the wrong screen, go to $O^{\varepsilon}_{45}$ |
| $O^{\varepsilon}_{49}$ | Hit ENTER twice and go to $O^{\varepsilon}_{51}$ |
| $\overset{17(1)}{\downarrow}\ O^{\varepsilon}_{50}$ | Type the Start Date and End Date (see pattern—MM/DD/YY), hit ENTER, and go to $O^{\varepsilon}51$ |
| $O^{\varepsilon}_{51}$ | Wait for the report completion message |

tions (th = thinking action category) before a decision is made. Performance of actions complying with these logical conditions is not altered by the improvements. Analysis of these logical conditions demonstrates that special training is required for efficient performance of the previously described members of the algorithm. In this case, training is not reducible to explanation and demonstration. Rather, a set of scenarios reflecting the diverse contingencies and outcomes must be developed around $l^{th}_6$ in combination with members $O^{\alpha}_{20}$, $O^{\varepsilon}_{21}$, and $O^{\varepsilon}_{23}$ of the algorithm. This shows that algorithmic formulation of a task is useful for training development.

Member of algorithm $O^{\alpha}_{26}$ is different before and after improvement. Before improvement, the message "PRICE DOES NOT MATCH INITIAL PRICE. Do You Wish To Proceed? (Y/N)" always emerges when the price is different than the price on the screen (order price). After improvement, this message appears only when a new price is more than 10% over order price. This reduces the perceptual workload during the performance of this member of the algorithm.

Members of the algorithm $O^{\alpha th}_{27}$, $l_9$, and $O^{th}\mu_{28}$ (see Table 1) are performed only before improvement. Taking into consideration that these members of the algorithm involve thinking and decision-making processes, their elimination is particularly potent by virtue of a reduction of task complexity. Logical condition $l_{10}$ before the improvement (Table 1) is carried out through maintenance of information in working memory until these logical conditions are completed. The function performed by a logical condition $l_{10}$ before improvement (see Table 1) is performed by $l_9$ after improvement (see Table 2). In this last case, decision making is carried out based on exteroceptive information presented on the screen. This significantly reduces the load on working memory and complexity of task performance in general.

Comparison of the methods of performance implicated in the evaluation of parts' price before and after improvement generally reveals that before improvement, multiple diverse steps including many behavioral and mental actions are required. After improvement, all of these actions were eliminated. There are also differences in the performance of logical condition $l_{10}$ (before improvement) and in $l_9$ (after improvement). After improvement, $l_9$ is performed not only on the basis of exteroceptive information but also partially automated by the computer system. The message appears on the screen only in those cases when price exceeds a threshold of 10% or more. In this case, the operator has a choice to answer "Yes" (Y) or "No" (N). In all other cases (price is less, equal, or less than 10% difference), the decision is made by the computer system. Only under special cases when a worker receives an instruction from his or her supervisor can the worker supply the answer "Yes" (Y), even when the price is more than 10% over order price. The system does not default to "N" if the price variance is more than 10% positive because under particular circumstances, it is possible for the worker to answer "Yes" (Y) if specific instructions are given by a supervisor. In this situation, a worker can hit "ENTER" prior to conscious decision making. Decision making connected with logical condition $l_9$ (Table 2) is only partially automated, providing for flexibility of worker performance and his or her ability to decide what to do in any particular case. Moreover, this prevents mindless hitting of "ENTER."

If a worker's answer is "No" (N), then he transfers to $O^{\varepsilon}_{27}$, $O^{\alpha}_{28}$ and $l_{10}$ (see Table 2). The worker then progresses to a new item or a new box with another item inside. By the same token, if it branches to ${}^{1}O^{\varepsilon}_{29}$, the cursor moves to the next field and the worker goes to $O^{\alpha}_{30}$. Members of the algorithm from $O^{\alpha}_{30}$ to $O^{\varepsilon}_{32}$ describe the worker's activity when he or she compared received and ordered quantities (see Table 2). This part of the task was not altered by intervention.

The following part of the algorithm describes WIP. This part of the task was already discussed previously. Consequently, the part of the algorithm that begins with $O^{\alpha}_{38}$ up to $O^{\varepsilon}_{40}$ (see Table 2) will be considered. In Table 1 (before improvement), this part of the task is described by members of the algorithm from $O^{\alpha}_{35}$ through logical conditions $l_{16}$. We compare these members of the algorithm with those mentioned in Table 2.

Under conditions in which the bin exists, the tasks performed before improvement and after improvement are the same. Before improvement, workers performed $O^{\varepsilon}_{40}$ following $l_{13}$ (see Table 1). Workers performed it the same way after improvement—$l_{14}$ and then $O^{\varepsilon}_{41}$ (see Table 2). However, in approximately 10% of the cases, bins are not assigned for particular items. Workers must categorize the item themselves. This part of the task is treated following. Prior to improvements and after $l_{13}$, workers should perform $O^{\mu}_{36}$ (see Table 1). Symbol μ designated the situation under which the worker must retrieve required information from long-term memory ("recall required bin type"; see $O^{\mu}_{36}$, Table 1) then follow $l^{\mu}_{14}$. There are different bin types. Based on the information retained in working memory, the worker makes a decision on which bin to select. Analysis of the previously mentioned members of the algorithm requires workers to continually maintain in memory required information, causing an overload on working memory. The decision-making process is based not on exteroceptive information but on information extracted from memory. This is a complicated decision-making process, which increases the probability of the computer system presenting a warning on the screen, "Wrong Bin Type." To avoid the warning, monitoring and control actions $O^{\alpha}_{38}$ through $l_{16}$ were introduced into the task algorithm, and if required, workers should return to $O^{\mu}_{36}$ (see Table 1).

After improvement, the list of bin types is presented on the screen (see $O^{\alpha}_{38}$, Table 2). This eliminates the necessity to retrieve information from long-term memory (instead of $O^{\mu}_{38}$ in Table 1, $O^{\alpha}_{38}$ in Table 2). Decision making is executed now on the basis of exteroceptive information presented on the screen rather than from information extracted from memory. In this case, mnemonic actions are transformed into perceptual actions. This facilitates the decision-making process and reduces the probability of an erroneous decision. As a result, members of the algorithm requiring control actions for correction can be eliminated. Moreover, if this correction is indicated, it can be performed with the assistance of information regarding the bin type from the screen.

The following members of this algorithm, $O^{\varepsilon}_{41}$ through $O^{\alpha}_{48}$ were not affected by improvements (see Table 2). We consider the final step of the algorithm of activity. This part of the algorithm describes those portions of the task that are involved in producing reports. A worker may print what he or she received during 1—up to several—days. In most cases, reports are related to what the worker did during the day. Before improvement, workers in both cases (report during 1 day or report during several days) must key in "Start Day," hit "ENTER" and key in "End Date" and hit "ENTER" (see $O^{\varepsilon}_{47}$, Table 1). Moreover, the computer system expects the date to be entered in a specific way. If the keyed in date pattern does not match the date pattern in the computer system, the report will be empty. Thus, it is important to provide the worker the date pattern so that he knows how to present the date to the system. This pattern was not presented before improvement.

After improvement, if a report is produced for 1 day, the worker simply hits "ENTER" twice (see $O^{\varepsilon}_{49}$, Table 2). If a report covers several days, the worker carries out $O^{\varepsilon}_{50}$ in Table 2, which corresponds to $O^{\varepsilon}_{47}$ in Table 1. The difference is that the worker is always presented with the required "MM/DD/YY" date pattern. This reduces the probability of errors caused by the worker's preferred pattern of keying in date fields. Consider also that workers produce reports for only 1 day, meaning that he or she typically only has to hit "ENTER" twice. In general, it may be seen that the suggested method of morphological description of activity that includes qualitative and algorithmic stages of analysis is a powerful tool to be used for the study of HCI.

Finally, we briefly discuss the cost-effectiveness of ergonomic intervention. Bedny, Karwowski, and Seglin (2001) considered this problem in a previous article pertaining to the analysis of physical labor performed under adverse working conditions. Our article here describes the analysis of a computer-based task in which the cognitive elements of work activity are of primary importance. To quantify and evaluate the cost-effectiveness of ergonomic intervention in a computer-based task, the researcher must evaluate the reduction of time expenditure yielded by the following ergonomic interventions:

1. Improvement in methods of presenting information on the screen.
2. Reduction in overall screen number.
3. Reduction in physical effort during performance of the computer based task.
4. Reduction in errors during task performance and reduction in time needed for error correction.
5. Reduction in mental and physical fatigue (even in the absence of work-time reductions, this element is important in the long term in that it decreases frequency of future illness and attrition in the work force).
6. Reduction in work-time of programmers involved in the design or redesign of the computer-based task (multiple redesign of the computer-based task can become extremely costly).

Elements 5 and 6 of a cost-effectiveness evaluation are particularly difficult to objectively quantify. The evaluation of these elements requires data collection and statistical comparison based on different aspects of work activity. In the example discussed in this study, the effectiveness of ergonomic intervention was evaluated based on an expert analysis, as a more detailed consideration of this topic was outside the scope of this project.

## 8. DISCUSSION

Activity is a complex, multidimensional system that calls for a systemic-structural method of study. In this discussion, rather than addressing a "human–machine system," activity itself is defined as a system. Such understanding of activity does not preclude the notion of a human–machine system or, as in our example, a human–computer system. To understand activity as a system, the formulation of basic units of activity analysis is necessary. Peculiarities of interrelations among these elements, what whole properties of the system are provided for, and what is the hierarchical organization of activity, and so forth must be comprehended. This work demonstrated the potential of this approach for HCI design.

In activity theory, a few discrepancies can be observed in some cases between the general principles of systemic-structural analysis and their realization. Accordingly, during the past 30 years, activity theory has dedicated substantial resources to formulate units of analysis (Bedny, 1987; Bedny et al., 2000; Gordeeva & Zinchenko, 1982; Zarakovsky & Pavlov, 1987). These units of analysis are important for development of a systemic-structural analysis of activity.

The systemic-structural analysis of activity in HCI can be described in terms of the morphological, functional, and parametric characteristics. Central to the proposed approach is the morphological method of study, with actions, operations, and members of the algorithm as major units of analysis. Note that this study is limited to only two stages of analysis—qualitative and algorithmic. In most cases, this is sufficient. When HCI is considered, the interaction between these two stages is not strictly sequential but iterative. A transition from one stage to another requires additional information regarding activity as an object of the study. It is possible to revisit preliminary stages of analysis with the advantage of additional insights obtained at the earlier stages.

In the description of a computer-based task with multiple approaches to satisfy the stated goals, probabilistic rather than deterministic methods for algorithmic description can be used. Many logical conditions have more than two outputs with different probabilities. Each member of an algorithm emerges as a subsystem of activity for examination. This is particularly important in the study of HCI because HCI typically allows multiple approaches and is intangible in much of its actions. Consequently, such activity cannot be not readily observed or described systematically. Computer users are often at a loss to describe their preferences and requirements. In such circumstances, programmers try to develop programs in the absence of an adequate comprehension of user needs. Users frequently formulate new needs or requirements on the fly. Moreover, after evaluating new systems, users often change their opinions. If the data have to be changed, not only the program at hand but other related programs may have to be updated as well. In most cases, testing of the systems must be redone. Software design becomes a blind trial-and-error process. The original design may have to be redesigned multiple times. To reduce the unnecessary intermittent steps, anticipation of the user needs through elaborate description of the computer-based task is vital.

The approach we proposed in this article can be very useful in eliciting from the prospective or actual users a precise description of their own criteria of performance. It also becomes possible to evaluate the efficiency of a computer-based task at earlier stages of design performance. The intermittent stages of design can thereby be significantly reduced, and design cycle times may be shortened.

Finally, we briefly compare the GOMS (goals, operators, method, selection) method (Card, Moran, & Newell, 1983) and SSTA. The two approaches are similar in that they both attempt to describe human activity on a computer-based task in an algorithmic or quasi-algorithmic manner. However, GOMS and SSTA differ in their methods for creating these descriptions. In SSTA, the algorithmic description of activity, although important, is not the only step in the analysis of activity. Rather, the algorithmic description stage is closely interrelated with other stages of the systemic-structural analysis. According to the GOMS approach, the algorithm of performance resembles a computer algorithm. In activity theory, one employs the concept of the human algorithm. The most important distinguishing feature of the human algorithm is the fact that the basic units of analysis are the cognitive/motor actions.

Within SSTA, researchers have developed methods for the extraction of different actions from the holistic process of activity. These actions are then classified according to principals defined by activity theory. SSTA proposes a hierarchical system of analysis units: operator, action, operation, and function block. These analysis units form a unified system, which allows for a standardized description of holistic activity. Any action directed to achieve a conscious intermediate goal should be distinguished from the final goal of a task. All actions have a loop structure. The starting point of any action is the moment when the goal is formulated or accepted. The endpoint of the action is the moment when the results of the action are evaluated. This permits a continuing flow of activity divided into individual units. Actions are considered as dynamic units that can be integrated into more complex actions or decomposed into smaller ones. This process depends on the complexity of the task and on the past experience of the performer. In SSTA, one can differentiate two kinds of units of analysis. One is called "typical elements of tasks" or "technological units," and the other is called "typical elements of activity" or "psychological units." The latter ones are standardized units of activity. Technological units can be transformed into psychological units at later stages of analysis. GOMS lacks strictly defined and unified units of analysis.

In SSTA, one can also distinguish deterministic and probabilistic algorithms that can be described in a symbolic manner. GOMS method does not consider these aspects of algorithmic description of activity. In view of SSTA, an activity is a multidimensional system that requires multiple approaches to the description of a single object and different stages of analysis. The distinctiveness of this analysis is that diverse descriptions of the same object must be interdependent and supplementary, enabling researchers to obtain a holistic picture of the object of study that capture various aspects of activity structure. In turn, the description of the structure of activity is compared with the physical and symbolic structure of the equipment. The GOMS method does not have standardized and interdependent stages of analysis.

SSTA, with its units of analysis, systemic-structural principles of human activity considered as a multidimensional system, and theoretically elaborated terminology, may be very useful in the study of HCI. However, although the SSTA was mainly developed as a psychological theory, it is not limited to the study of HCI only. This theory can be applied to study of different kinds of human work, including training and education in general.

## REFERENCES

Bannon, L., & Bodker, S. (1991). Beyond the interface: Encountering artifacts in use. In J. Carrol (Ed.), *Designing interaction: Psychology of human computer interface* (pp. 227–253). Cambridge, England: Cambridge University Press.

Bedny, G. Z. (1987). *The psychological foundations of analyzing and design work processes.* Kiev, Ukraine: Higher Educational Publishers.

Bedny, G. Z. (2000). Activity theory. In W. Karwowski (Ed.), *International encyclopedia of ergonomics and human factors* (Vol. 1, pp. 358–362). London: Taylor & Francis.

Bedny, G. Z., & Karwowski, W. (2000). Theoretical and experimental approaches in ergonomic design: Towards a unified theory of ergonomic design. *Proceedings of the XIVth Triennial IEA Congress, and the 44th Annual Meeting of The Human Factors and Ergonomics Society, 5,* 197–200.

Bedny, G., Karwowski, W., & Bedny, M. (2001). The principle of unity of cognition and behavior: Implications of activity theory for the study of human work. *International Journal of Cognitive Ergonomics, 4,* 401–420.

Bedny, G. Z., Karwowski, W., & Jeng, O.-J. (2001). Activity theory as a framework to study human–computer interaction. *International Conference on Computer-Aided Ergonomics and Safety,* Maui, Hawaii, 1–4.

Bedny, G. Z., Karwowski, W., & Kwon, Y.-G. (2001). A methodology for systemic-structural analysis and design of manual-based manufacturing operations. *Human Factors and Ergonomics in Manufacturing, 11,* 233–253.

Bedny, G. Z., Karwowski, W., & Seglin, M. (2001). A heart rate evaluation approach to determine cost-effectiveness an ergonomics interventions. *International Journal of Occupational Safety and Ergonomics, 2,* 121–133.

Bedny, G. Z., & Meister, D. (1997). *The Russian theory of activity: Current applications to designing and learning.* Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Bedny, G. Z., & Meister, D. (1999). Theory of activity and situation awareness. *International Journal of Cognitive Ergonomics, 1,* 63–72.

Bedny, G. Z., & Seglin, M. H. (1999). Individual style of activity and adaptation to standard performance requirements. *Human Performance, 1,* 59–78.

Bedny, G. Z., Seglin, M. H., & Meister, D. (2000). Activity theory: History, research and application. *Theoretical Issues in Ergonomics Science, 2,* 168–206.

Brushlinsky, A. V. (1979). *Thinking and forecasting.* Moscow: Science Publishers.

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

Decortis, F. (2000). Activity theory, cognitive ergonomics and distributed cognition: Three views of a transport company. *International Journal of Human–Computer Studies, 53,* 5–33.

Engestrom, Y., Miettinen, R., & Punamaki, R.-L. (1999). *Perspectives on activity theory.* New York: Cambridge University Press.

Gordeeva, N. D., & Zinchenko, V. P. (1982). *Functional structure of actions.* Moscow: Moscow University Publishers.

Hacker, W. (1985). Activity: A fruitful concept in industrial psychology. In M. Frese & J. Sabini (Eds.), *Goal-directed behavior: The concept of action in psychology* (pp. 81–98). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Kuutti, K. (1996). Activity theory as a potential framework for human–computer interactions. In B. A. Nardi (Ed.), *Context and consciousness: Activity theory and human–computer interaction* (pp. 17–44). Cambridge, MA: MIT Press.

Leont'ev, A. N. (1977). *Activity, consciousness and personality*. Moscow: Political Publishers.

Nardi, B. A. (Ed.). (1997). *Context and consciousness: Activity theory and human–computer interaction.* Cambridge, MA: MIT Press.

Rubinshtein, S. L. (1973). *Problems of general psychology.* Moscow: Academic Science.

Volpert, W. (1982). The model of the hierarchical-sequential organization of actions. In W. Hacker, W. Volpert, & M. Cranach (Eds.), *Cognitive and motivational aspects of action*. Berlin, Germany: Huthing Velagsgemeinschaft GmbH.

Vygotsky, L. S. (1978). *Mind in society. The development of higher psychological processes.* Cambridge, MA: Harvard University Press.

Zarakovsky, G. M., Korolev, B. A., Medvedev, V. I., & Shlain, P. Y. (1974). *Introduction to ergonomics.* Moscow: Soviet Radio.

Zarakovsky, G. M., & Pavlov, V. V. (1987). *Laws of functioning of man–machine systems.* Moscow: Science Publishers.