UDC:159.98
DOI: https://doi.org/10.24195/2414-4665-2019-3-8

Inna Bedny,
Doctor of Psychological Sciences, professor,
Information Technology, UPS,
3, Nuthatcher Court, Wayne, New Jersey, USA

HUMAN FACTOR IN PERFORMING COMPUTER BASED AND COMPUTERIZED TASKS: SYSTEMIC-STRUCTURAL ACTIVITY THEORY PERSPECTIVE

The purpose of this article is to highlight the issues related to human-computer interaction (HCI) and demonstrate how application of the Systemic-Structural Activity Theory (SSAT) allows to solve some of the problems related to utilization of software. Wide spread HCI is a relatively new phenomenon. Today people of all ages use gadgets to connect with their friends and relatives, to purchase all kinds of necessities, to work, travel and play. Software developers are often concentrated on coding while ignoring the human element. The software intended for work that is used hundreds of times during the day should be designed differently from the interface for the customers that might have to use it once. SSAT considers human activity as a structure that unfolds in time. Suggested in this framework methods of analysis of human performance allows to analyze the structure of the task, find its critical points, determine the complexity of task performance and compare different versions of interface design to choose the best version. SSAT suggests new methods of algorithmic description of human performance that allow to conduct the qualitative analysis of the software used to perform the task and derive the quantitative measures of complexity. Special attention is paid to the decision-making process and to the reliability of human performance. The developed analytical methods can be used at the design stage when software does not exist yet. These methods help to choose the best design version saving a lot of resources at the early stages of the design process. Here we are going to demonstrate how applying these methods we can improve user experience by make the task performance less complex and more efficient.

Keywords: *user experience, task analysis, design, complexity, reliability of human performance*.

Introduction

Interaction of people with various software is the relatively new phenomenon. Personal computers became a new reality about 25 years ago. The generation that grew up within this timeframe always find their way around in any software. Older people have harder time to use new applications.

Optimization of software design is especially important for websites of the businesses. The clients have very little time to find what they are looking for and if it takes too long, they find a more user-friendly website of the competitors. This issue is analyzed in our paper dedicated to the abandoned actions (Bedny, 2011).

The software that is utilized at the workplace has to be optimized to make the performance efficient, the demands of keeping information in working memory should be minimized.

Developed by Gregory Bedny Systemic-Structural Activity Theory (SSAT) is a framework that offers a battery of methods for analysis and assessment of human performance. These methods can be utilized at the design stage or applied to the evaluation of enhancements and innovations, and to the comparison of different versions of software, equipment or task performance.

Changes in the software or equipment configuration lead to changes in the strategies of task performance. Comparing such configuration with the structure of human activity should be the basic principle when designing human-computer interaction (HCI) or human-machine systems (Bedny & Bedny, 2018).

Methods that allow to predict the validity of future enhancements and to evaluate the efficiency of the model at the design stage allow to reduce the number of design-development-implementation cycles saving a lot of efforts and resources.

Task analysis is central, because systems with inadequate functionality frustrate the users and are often rejected or underutilized (Shneiderman, 1998).

Purpose

In this paper we demonstrate the new method of human computer interaction task analysis. The suggested method presents human algorithm of task performance in a standardized manner. Such algorithm gives a precise description of what is involved in completing a task at hand. Algorithmic description is accompanied by the time structure analysis. Knowing the time, it takes to perform each member of the human algorithm and probability of its occurrence makes quantitative analysis of human performance possible.

The qualitative and quantitative analysis of such algorithm allows to improve the design of a new non-existing interfaces or an interface that is already in use. In a design of any complex artefact a range of representations, or models, is needed during the design process (Preese, 1994).

The qualitative analysis of task performance of the same task utilizing different versions of software allows to assess complexity of human activity. Such evaluation is of vital importance for evaluation of efficiency and reliability of performance. According to Simon (Preese, 1994), complexity is the basic property of a system. Human activity is a complex structure that unfolds in time (Simon, 1999). The algorithmic description of this structure accompanied by the time structure and probability analysis that depicts the flow of the variable human activity allows to determine the complexity of tasks.

SSAT offers multiple measures of complexity evaluation.

Reducing the task complexity allows to increase productivity and to improve the user experience.

Method

A. Morphological Analysis of Activity

Morphological analysis was developed by Zwicky (Zwicky, 1969) as an independent approach to the study of complex systems in an abstract manner. In SSAT this method is utilized for the description of the structure of human activity during task performance. This approach employs the standardized language of activity description that allows development of analytical models of activity. SSAT is the unique approach that can create analytical model of extremely variable activity. This morphological analysis includes two stages: algorithmic task description and time structure analysis.

B. Algorithmic Description of Activity

A human algorithm is a system of logically organized mental and motor actions that is aimed at solving a specific class of problems or at performing various tasks.

The basic units of such human algorithm are symbols that identify cognitive and behavioral actions. This algorithm depicts logically organized elements of human activity verbally and symbolically.

Table 1

Algorithmic Description of Activity and its Time Structure during Computerized Task Performance

Members of Algorithm (psychological units of analysis)	**Description of Elements of task (Technological units of analysis)**	**Description of Elements of activity (Psychological Units of Analysis)**	**Time** sec
O^{α}_{1}	Check for presence of inventory receiving screen	Simultaneous perceptual action (ET + EF)	0.42 + 0.3 = 0.72
$\overset{1}{\downarrow} O^{\varepsilon}_{2}$	Type 1 and press ENTER to choose ADD INVENTORY RECEIVING screen	(R50B+AP1) + (R30B + AP1)	1.68 × 1.2 = 2.01
— — —	— — — — — — — — —	— — — — — —	— —
O^{ε}_{13}	Press ENTER to go to the screen with detailed item information	Motor action (R26B + AP1)	0.76×1.2 = =0.9
$O^{\alpha\mu}_{14}$	Compare received quantity with PO (purchase order) quantity	Combination of two simultaneous perceptual actions - 2 × (ET + *EF*) with simultaneously performed mnemonic operation (MO)	(0.42 + *0.4*) × 2= =1.64
$l_5 \overset{5}{\uparrow}$	If received quantity and order quantity are the same, go to O^{ε}_{22} (P=0.9). If received quantity is greater or less than order quantity, go to O^{ε}_{15} (P=0.1)	Decision-making action performed based on visual information	0.4
O^{ε}_{15}	Type received quantity and press ENTER to get a question at the bottom of the screen (P= 0.1)	Motor action (R20B + AP1) + (R12B + AP1) (example with two digits number)	(0.8 × 1.2) × 0.1 = =0.096
O^{α}_{16}	Read the statement: THE RECIVED QUANTITY AND ORDER QUANTITY DO NOT MATCH. DO YOU ACCEPT? (YES/NO). (P = 0.1). Scan and read about four words.	Successive perceptual action. ET + 4 × *EF*	(0.42 + 4 × 0.18) = =1.14×0.1= 0.11

Table 1 consists of four columns. The first column on the left depicts the symbolic description of the members of this human algorithm, the second column represents elements of the task as technological units of analysis, the third column consists of description of the psychological units of activity and the last column gives the time it takes to perform each element of this task.

Members of the algorithm are of two basic types: operators (*O*) and logical conditions (*l*). Operators represent motor (O^{ε}) or cognitive actions (O^{α}) that correspond to transformation of objects, energy, or information while logical conditions reflect decision-making actions. Symbol $O^{\alpha\mu}$ shows that information had to be recalled or kept in working memory. The decision-making actions determine the flow of the algorithm. Tables 1 is the examples of the algorithmic description of task performance.

C. QUANTITATIVE ASSESEMENT OF TASK COMPLEXITY

Presented above human algorithm depicts task performance in a standardized manner. It gives a precise description of what is involved in completing a task at hand and allows to improve the design of new or existing interfaces.

Assessment of complexity of human activity that is considered as a system is of vital importance for evaluation of efficiency and reliability of performance. According to Simon (Preese, 1994), complexity is the basic property of a system. Human activity is a complex structure that unfolds in time (Simon, 1999). The algorithmic description of this structure accompanied by the time structure and probability analysis that depicts the flow of the variable human activity allows to determine the complexity of tasks.

Algorithmic description is accompanied by the time structure analysis. Knowing the time, it takes to perform each member of the human algorithm and probability of its occurrence makes quantitative analysis of human performance possible.

SSAT offers multiple measures of complexity evaluation. Let us present some of them as an example.

The duration of perceptual components of task is calculated as:

$$T\alpha = \Sigma P^{\alpha} t^{\alpha} \quad (1)$$

The duration of thinking components of task is:

$$Tth = \Sigma P^{th} t^{th} \quad (2)$$

The time spent on retaining information in working memory is calculated using the following formula:

$$Twm = \Sigma P_{wm}\, t_{wm} \quad (3)$$

And the duration of decision-making components of task is defined as:

$$L_g = \Sigma P^l\, t^l \quad (4)$$

In the above presented formulas P is the probability of the occurrence of the corresponding member of the algorithm and t is its duration.

The duration of all cognitive components of the task are determined by utilizing the formula below:

$$Tcog = T\alpha + Tth + Twm + L_g \ (5)$$

Other absolute measures such as time spent on the executive components of task, total time for task performance, and so on are determined similarly. SSAT also offers relative measures of complexity. For instance, fraction of time spent on thinking operators in the entire time of task performance is determined as:

$$Nth = Tth / T \ (6)$$

In formula (6) T is the total time of the task execution.

We can also determine the percentage of time spent on keeping information in working memory, on decision-making, and so on (Bedny, 2019).

Each of these measures describes what is involved in a particular task performance. The specificity of the task under consideration can determine which measures are the most important ones for its analysis. The described approach allows to create new measures of complexity if necessary.

In case there is a need to evaluate the innovation the algorithmic description of the task performance before and after innovation should be developed along with the time structure analysis. As the next step, the absolute and relative measures of complexity that are the most important ones for the given task should be calculated. Comparison of these measures such as total time for the task performance, percentage of time spent on decision-making, length of the waiting periods, and so on, before and after its implementation would depict a true value of the proposed innovation.

Analysis of these measures and their comparison gives a true picture of the task complexity and identifies the root cause of the issues related to the performance of the task that might include overload on the short-term memory or require numerous decision-making in a short period of time. Without such thorough analysis it is hard to uncover the design issues because two task performances can look very similar to the observer while one of them might include much higher cognitive workload.

Determining complexity of task performance demonstrates cognitive demands on human activity and helps to optimize it. Optimization of task performance enhances efficiency of performance, reduces errors and decreases the probability of system failures. Complexity of task performance is tritely connected with its reliability.

Findings

SSAT can be instrumental for the error analysis and for determining the causes of the system failures. It offers a new event tree modeling method for evaluation of the probability of the successful performance or failure. Figure 1 depicts the event tree of the item quantity evaluation described in Table 1. This event tree just reflects the fragment of the task performance.

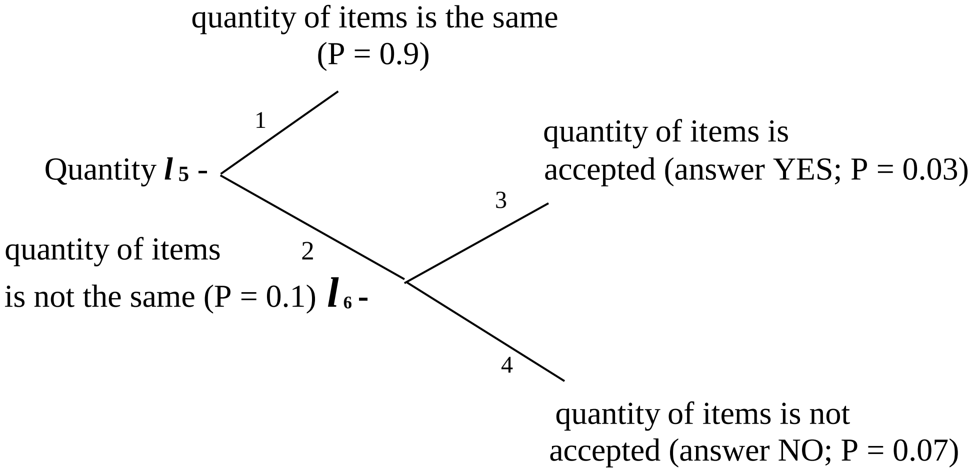

Figure 1. Event-tree of the item quantity is evaluation

Table 1 and Figure 1 are the human algorithm and the event tree fragments of the analysis of the computerized task of receiving inventory in the warehouse.

Computerized tasks are the tasks that combine physical elements and utilization of a software. Logical conditions are critical points where the probability of performance of various elements of the task are determined. The suggested method of event tree model development helps experts to determine probabilities of events with the high precision. The event tree model can also simplify the calculations.

Let's as consider how various decision-makings affect the probabilistic structure of the human algorithm. We want to bring readers' attention to the fact that in traditional event-tree models, each branch is considered as an independent one and has probability from zero to one. In our model, the probability of preceding branches determines the probability of the following branches.

Figure 1 demonstrates that the outcome of each branch is clearly associated with decision-making actions performed by an operator. The event-tree utilizes the probabilities that were obtained by estimations provided by the subject matter experts and by calculations. Letter P on each figure identifies the probability of the decision-making outcomes. It also demonstrates probabilities of failure and successful performance.

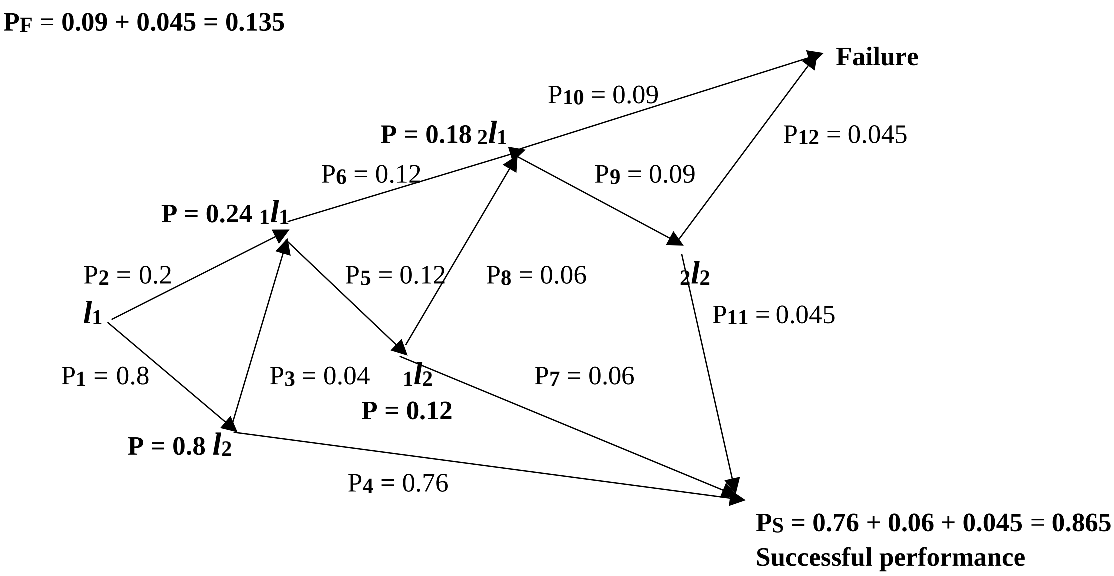

Figure 2. Event-tree model of the task performance utilizing the old version of the software

The event-tree method is used in combination with morphological analysis of task execution. It presents all decision-making actions (logical conditions) performed by an operator in a concise form.

The event-tree depicts the relationship between logical conditions (decisions makings) that determine the probabilistic structure of the considered task. Logical conditions and their probabilities are depicted by the dots that connect the lines.

Outcomes of logical conditions (decision makings) are depicted by lines with their probabilities. Figure 1 shows how outcomes of decision-making influences probabilities of events and the reliability of the task performance. The meaning of each outcome in the presented event tree is described in Table 1. Its full version can be found in G. Z. Bedny and I. S. Bedny (Bedny & Bedny, 2019).

Figures 3 and 4 demonstrate application of the event tree model for analysis of the computer-based task of transferring the file from one server to another in the industrial settings. The computer-based tasks are completed solely by using the software. Creating the event-trees of the whole task before and after improvement of the software allows to demonstrate how implementation of the innovation increased the probability of the successful performance of the task.

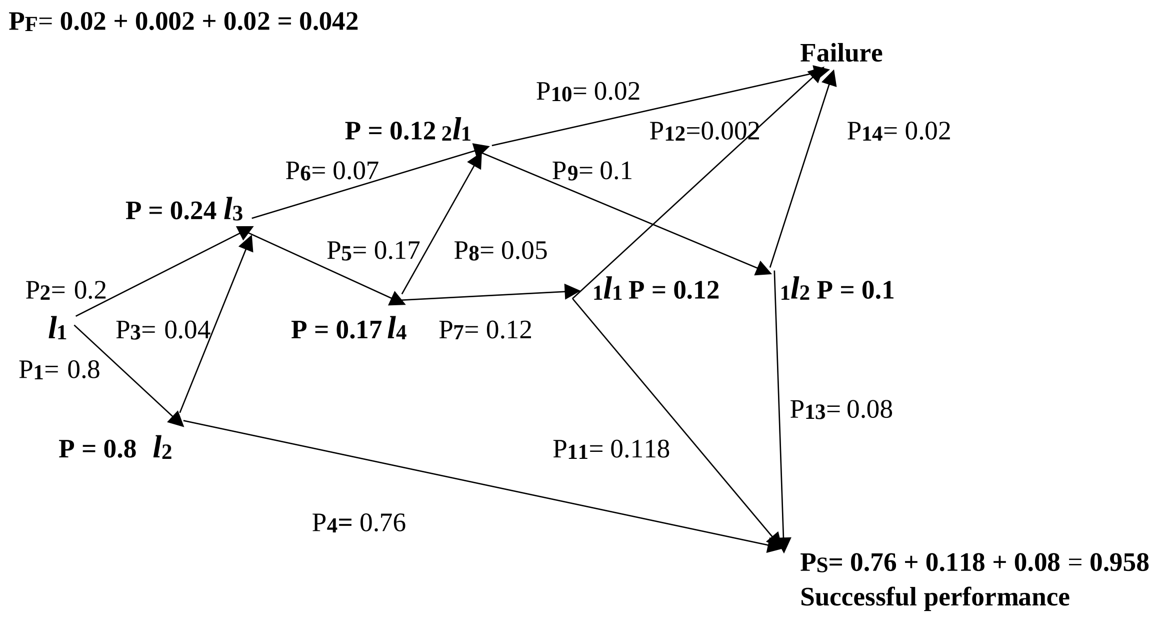

Figure 3. Event-tree model of the task performance utilizing the enhanced software

Figure 3 and 4 demonstrates a new event tree modeling method for evaluation of the probability of successful performance and the probability of failure.

Let us consider the event-tree depicted on Figure 3. The first logical condition (l_1) shows the decision when the file name "orders" is browsed for on the list of file names. This logical condition has two outcomes. The first outcome has probability $P_2 = 0.2$ when the file is not found on the list. The second outcome has probability $P_1 = 0.8$ and demonstrates the situation when the file name is found on the list.

If the file name is not on the list, the operator needs to restore communication. If the file is on the list, the operator has to check the date stamp of the file. This involves the second decision designated by logical condition l_2. This logical condition also has two outcomes. If the file has the date different from the current date (P3 = 0.04), the operator needs to restart communication. This outcome of l_2 (P3 = 0.04) converges with output of l_1 (P2 = 0.2).

Thus, these probabilities are combined for the following logical condition ${}_1l_1$ that has the probability P = 0.2 + 0.04 = 0.24. Typically, the event-tree shows a situation when branches of events only diverge, but in this practical situations, branches of events in some cases converge (see as an example ${}_1l_1$) and the probabilities of the events are summarized. The logical condition ${}_1l_1$ has two outcomes with equal probabilities of P=0.12 (P5 and P6). One of these outcomes leads to the next logical condition ${}_2l_1$ with the probability of 0.12.

The second outcome leads to the next logical condition ${}_1l_2$ that has the probability of 0.12. It also has two outcomes with equal probabilities of 0.06. One of these outputs also leads to the logical condition ${}_2l_1$. As a result, the logical condition ${}_2l_1$ has a probability of P = 0.12 + 0.06 = 0.18. The other probabilities can be described similarly. The probability of failure is a combination of outcomes of two logical conditions ${}_2l_1$ and ${}_2l_2$, and therefore $P_F = 0.09 + 0.045 = 0.135$. The probability of the successful result is a combination of outcomes of l_2, ${}_1l_2$, and ${}_2l_2$, and is calculated as $P_S = 0.76 + 0.06 + 0.045 = 0.865$.

Finally, it should be pointed out that decisions (logical conditions) can have more than two outcomes with various probabilities. At the first stage, experts can use the developed in SSAT scale of the subjective probability evaluation of events. The event tree helps to improve the accuracy of determining the probabilistic structure of activity during task performance. The algorithmic and time-structure description of the existing method of task performance demonstrates that the probabilistic structure of activity depends on outcomes of logical conditions (decision-making actions).

The presented above event-tree model demonstrates that it is useful for evaluating the probability of failure and of successful performance. If we compare the probabilities of the success and failure of the task performance before and after implementation of the enhancement of the software we can see from Figures 3 and 4 that before the improvement the probability of success was $P_S = 0.865$ and after the improvement it was $P_S = 0.958$ (see Figure 4) and the probabilities of failure were $P_F = 0.135$ and $P_F = 0.042$.

Such analysis clearly demonstrates that the implemented enhancement improved the chances for the successful performance of the task.

It is also beneficial to use the graphical form of the description of the activity algorithm along with the event tree model method (Bedny, 2018). Such combination of methods can provide a clear picture of the efficiency of the task performance and of its reliability.

The event-tree model helps to visualize probabilities of transition from one member of the algorithm to the next and to verify the already obtained data. Our enhanced method of determining probabilities, that applies the subject matter experts' and calculated probabilities in combination with the event tree method, makes the analytical part of task analysis more efficient.

This approach eliminates the necessity of obtaining the experimental data that is often hard to collect and expensive to collect.

Discussion

The method described in this paper allows to estimate with high precision the reliability of task performance. For the first time, it's demonstrated how the new method of event tree development can be used for reliability assessment of user performance.

Furthermore, this method allows to assess with high accuracy the complexity of tasks that has a complex probabilistic structure.

The combination of various methods of task analysis is very useful for the refinement of the algorithmic descriptions of activity.

This method of task analysis allows assessment of the efficiency of the innovations aimed at improvement of the reliability of performance.

Instead of implementing the innovation that might be rather costly and inefficient, application of this method that includes reliability and efficiency analysis helps to determine if the propose innovation going to be cost-efficient (Bedny, 2019).

Complexity and reliability are interdependent characteristics of task performance. When complexity increases it often leads to decrease in the reliability of performance.

Conclusion

With increased cognitive demands to task performance, psychological methods of studying human activity play an important role. SSAT is a comprehensive unified psychological theory that can be utilized as a general approach to the study of human activity.

The developed in the framework of SSAT approach to the study of highly variable human activity allows to create efficient methods of its analysis. Human activity is considered as a self-regulative system that unfolds in time.

The described in this paper analytical methods just give a glimpse of possibilities that this approach opens. It can be applied to enhancement of the existing tasks, interfaces and equipment as well as to the design of the new tasks and processes.

This framework can be also utilized to evaluation of innovations, comparison of different versions of design, analysis of human activity that is going to be replace with bots or artificial intelligence, and so on.

Application of SSAT leads to reduction of human errors, improvement of user experience, and efficiency of performance. This approach can be used in various fields.

The suggested by SSAT analytical principles of task analysis allow to reduce costly cycles of the continues enhancements of production processes and repetitive improvement of user interfaces. This approach facilitates analysis of extremely variable human activity and identifies the preferable strategies of task performance.

Systemic rigorous description of activity structure and evaluation of its efficiency is achieved through morphological analysis that facilitates the quantitative analysis of human performance.

SSAT offers quantitative methods for assessing psychological complexity and reliability of human performance that can save time and money.

Systemic description of the activity structure and evaluation of its efficiency are achieved via morphological analysis, which facilitates development of quantitative methods of analysis. It is vitally important to estimate task complexity because it determines the cognitive demands for its performance. The described methods allow to evaluate task complexity with high precision.

The concept of task and job complexity is important in economic analysis of the relationship between productivity and human performance. The more complex the task is, the greater is its cost per unit of time. Hence, this is a point of tight interaction between economics, psychology, and ergonomics.

The functional analysis of activity is of particular importance, when a complex self-regulative system is under consideration.

The analytical principles of task analysis allow to reduce costly cycles of continuous enhancement and redesign solutions of production processes, and the repetitive improvements of the software.

In general, this paper presents the advanced approach to the study of human performance from the SSAT perspective. It demonstrates application of this approach using extremely complex computer-based and computerized tasks.

REFERENCES

1. Bedny, G. Z., Bedny, I. S. (2018). "Applied and Systemic-Structural Activity Theories", Boca Raton, Florida, USA, CRC Press, Taylor and Francis Group [in English].
2. Bedny, G. Z., Bedny, I. S. (2019). "Work Activity Studies Within the Framework of Ergonomics, Psychology, and Economics", Boca Raton, Florida, USA, CRC Press, Taylor and Francis Group [in English].
3. Bedny, G. Z., Karwowski, W., Bedny, I. S. (2018). "Applying Systemic-Structural Activity Theory to Design of Human-Computer Interaction Systems", Boca Raton, Florida, USA, CRC Press, Taylor and Francis Group [in English].
4. Bedny, I. S. (2019). Application of systemic-structural activity theory to evaluation of innovations. Washington, DC, AHFE [in English].
5. Bedny, I. S., Bedny, G. (2011). Analysis of abandoned actions in the email distributed tasks performance. In D. B. Kaber, G. Boy (Eds.). *Advances in cognitive ergonomics*. (pp. 683-692). Boca Raton, London: CRC Press, Taylor & Francis Group [in English].
6. Bedny, I. S., (2019). Application of Systemic-Structural Activity Theory to Evaluation of Innovations. In: Janusz Kacprzyk (eds.) Advances in Neuroergonomics and Cognitive Engineering (Advances in Intelligent Systems and Computing), vol. 953, Springer International Publishing [in English].
7. Preese, J. (1994). "Human-Computer Interaction", Harlow, England, Addison-Wesley [in English].
8. Shneiderman, B. (1998). "Designing the User Interface", 3rd ed., Reading, Massachusetts, USA, Addison Wesley Longman, Inc. [in English].
9. Simon, H. (1999). "The Science of the Artificial", 3rd ed., Cambridge, Massachsette, USA, MIT Press [in English].
10. Zwicky, F. (1969). Discovering, Invention, Research – Through the Morphological Approach, Toronto, ON, Canada, The Macmillian Company [in English].

Інна Бедний,

доктор психологічних наук, професор,

Інформаційні технології,

Американська виробничо-комерційна поштова служба,

вул. Нотхече Корт, 3, Уейн, Нью-Джерсі, США

ЛЮДСЬКИЙ ФАКТОР У ВИКОНАННІ КОМП'ЮТЕРНИХ ЗАВДАНЬ: ПЕРСПЕКТИВА СИСТЕМНО-СТРУКТУРНОЇ ТЕОРІЇ ДІЯЛЬНОСТІ

Метою даної статті є висвітлення проблем, пов'язаних із взаємодією людини з комп'ютером (HCI), і демонстрація того, як застосування системно-структурної теорії діяльності (SSAT) дозволяє вирішити деякі проблеми, пов'язані з використанням програмного забезпечення. Широке поширення HCI є відносно новим явищем. Сьогодні люди різного віку використовують гаджети, щоб спілкуватися зі своїми друзями і родичами, купувати все необхідне, працювати, подорожувати і грати. Розробники програмного забезпечення часто концентруються на кодуванні, ігноруючи людський фактор. Програмне забезпечення, призначене для роботи, яке використовується сотні разів протягом дня, має бути розроблено інакше, ніж інтерфейс для клієнтів, яким, можливо, доведеться використовувати його один раз. SSAT розглядає діяльність людини як структуру, яка розгортається в часі. Запропоновані в цій теорії методи аналізу продуктивності людини дозволяють аналізувати структуру завдання, знаходити її критичні точки, визначати складність виконання завдання і порівнювати різні варіанти дизайну інтерфейсу, щоб вибрати кращу версію. SSAT пропонує нові методи алгоритмічного опису продуктивності людини, які дозволяють проводити якісний аналіз програмного забезпечення, використовуваного для виконання завдання, і виводити кількісні показники складності. Особлива увага приділяється процесу прийняття рішень і надійності роботи людини. Розроблені аналітичні методи можуть бути використані на етапі проектування, коли програмне забезпечення ще не існує. Ці методи допомагають вибрати кращу версію дизайну, економлячи багато ресурсів на ранніх етапах процесу проектування. Тут ми збираємося продемонструвати, як за допомогою цих методів ми можемо поліпшити взаємодію з користувачем, зробивши виконання завдання менш складним і більш ефективним.

Ключові слова: досвід користувача, аналіз завдань, проектування, складність, надійність працездатності людини.

Подано до редакції 10.12.2019