INNOVATIVE MECHANICAL ENGINEERING **ISSN 2812-9229 (Online)**
University of Niš, Faculty of Mechanical Engineering
VOL. 1, NO 2, 2022, PP. 80 - 95

Original scientific paper *

CAD SYSTEM EVALUATION BASED ON USER INTERFACE EFFICIENCY

Dragan Mišić[1], Miroslav Trajanović[1], Nikola Korunović[1], Jovan Aranđelović[1], Pavle Drašković[1] and Rajko Turudija[1]

[1]Faculty of Mechanical Engineering, University of Niš, Serbia

Abstract. *Differences between contemporary CAD systems made by different manufacturers are mainly reflected in their user interface. In this paper, we describe a method developed for an objective assessment of user interface efficiency. The method is based on calculating the active time of using a system, which is inversely proportional to efficiency. Active time is defined as the product of the number of elementary actions (mouse click, keyboard input, etc.) and the average duration of those actions. The average duration of actions was determined by statistical methods, using data collected from the application that we specifically developed for this purpose. Using active time as a measure, the evaluation of the interface is accomplished in an objective manner, since the influence of user skills is eliminated. We demonstrated the applicability of our method by testing four commercial CAD systems. The results can be used as one of the criteria for choosing one of the CAD systems, but they can also be a useful guide for improvements in interface design. The method described in this paper can be used to evaluate the efficiency of user interface of any computer system.*

Key words: *CAD system evaluation, user interface efficiency assessment*

1. INTRODUCTION

Choosing the right Computer-Aided Design (CAD) system today is not an easy task. There are a lot of products on the market which have very similar capabilities. If one product introduces a new feature, it is very soon available in other products as well. This all leads to choosing the CAD system based on personal affinities or price, as the most important factors.

To be able to choose the right system, there has to be a way to compare the quality of these systems. There are different aspects to account for when assessing software products, but they all rely on the usability which is crucial to determine the software quality.

***Received: December 03, 2021 / Accepted Match 23, 2022.**
Corresponding author: Dragan Mišić
Faculty of Mechanical Engineering, University of Niš
E-mail: dragan.misic@masfak.ni.ac.rs

[1] defines usability based on (i) effectiveness, (ii) efficiency and (iii) satisfaction in the given usage context. According to the same standard, effectiveness can be defined as accuracy and completeness with which users achieve specified goals, efficiency refers to the resources used to achieve the results (time, human effort, costs, and materials), and satisfaction can be defined as the extent to which the user's physical, cognitive, and emotional responses that result from the use of a system, product or service meet the user's needs and expectations.

Usability defined in this manner is often a distinguishing feature of software systems. To assess usability different evaluation methods are used.

According to [2], there are three basic types of usability evaluation methods. Those are testing, inspection and inquiry. Usability testing captures usage data from real end users while they are using the product to complete a predefined set of tasks. Usability inspection refers to interface evaluation performed by experts from the domain being tested. The assessment of the usability of the software depends on the person performing the testing, which means that the assessment is subjective. One of the ways to gather impressions about the software is to talk to users. In the inquiry method, the information about the system is obtained from conversations with users or observation of their work under real conditions.

The data obtained by evaluation can be quantitative and qualitative [3, 4]. Quantitative data are numerical data usually gathered by measurement, while qualitative data are textual descriptions of problems. Quantitative data most often refer to the time that the user needs to complete a specific task, the number of tasks that can be performed in a given interval, the number of errors, the relationship between successful task execution and the time required to complete the task, input speed, etc. [5, 6]. In our research, we also rely on the time of task execution, but in our case it is obtained in a more objective way, as the duration of direct interaction with the system.

One of the problems with most existing usability evaluation methods is that the evaluation result is a user's subjective assessment. Even when the evaluation is performed on the basis of a quantitative characteristic, such as the task execution time, that time depends, among other things, on the user's capabilities and his current state. Our method attempts to eliminate this subjectivity from the evaluation by concentrating on monitoring the actual interaction with the program and evaluating only the efficiency of the user interface.

Our method is similar to the Keystroke-Level Model (KLM) [7, 8], which is based on counting keystrokes and other low-level operations. In addition to these operations, this model also monitors the mental preparation of the user, as well as the response time of the system.

The difference between the basic KLM model and our model is that our model has a different set of basic operators (for example, we specifically monitor the selection of the left, right and middle mouse buttons, and the real mouse path). Also, in our model the average time needed to complete an elementary operation is determined empirically, on the basis of a large sample. In addition, we do not monitor mental effort, since it is a subjective characteristic, nor the response time of the system, since it is no longer relevant with modern computers and programs.

Most of the existing studies related to usability testing deal with web and mobile applications, which is to be expected since these applications are the most numerous. In [9] over 5000 studies that analyze web applications were found, while [10] mentions close to 800 documents analyzing usability in mobile applications.

According to [11], there are very few studies which examine CAD systems from the user interface point of view. The main reasons behind a small number of CAD system usability studies are (i) the complexity of those systems which requires domain experts to perform the evaluation and (ii) the smaller number of beneficiaries of the results since the engineering community is much smaller than the community of web users where this type of testing is much more common [12].

In contrast, the ergonomic aspect of CAD systems is very important for efficient work. According to [13], neglecting the ergonomic rules leads to reduced efficacy of the design process, decreased efficiency and a more significant strain for the designers.

The existing research concerning usability analysis in CAD systems is mainly based on the observation of the work of engineers or interviews with them. For instance, in [14] the author tracked the manual activities of engineers during the design process in order to improve CAD systems. The activities were tracked via video recordings which were later analyzed. In another study [15], the entire work of the user was automatically recorded in log files. Later, these files were analyzed in order to improve the system.

Architects also took an interest in CAD system usability testing. One of the studies in CAD systems applications in the architecture analyzed 10 CAD systems and tracked the errors during system usage, conducted questionnaires with trainees after the training sessions of the system and held focused group interviews with trainers [16]. Based on this data, they identified 10 principles related to user interface which can be applied to complex 3D systems. Those principles are Consistency, Visibility, Feedback, Recoverability, Maximization of Workspace, Graphical Richness, Direct Manipulation, Familiarity, Customizability, Assistance, Minimalist Design, and Context Recognition.

Another research related to CAD systems for architecture analyzed the applications of physiological and subjective measures in assessing the usability, effectiveness, and efficiency of the user interface [17]. Their goal was to reduce the usage of cognitive resources for controlling or understanding the interface.

In this paper, we show how CAD systems can be compared based on the efficiency of their user interface. For the comparison we use the efficiency-based method allowing for objective assessment, partially described in our previous work [18, 19]. Out of the three usability aspects described before, we focus on efficiency since we find it allows for objective assessment.

Our method consists of tracking elementary actions performed by the user while working with the program and the quantification of those actions. The elementary actions are performed through graphical user interface (GUI) and include mouse clicks, keyboard entries, mouse movements, as well as switching (moving hand) between the mouse and the keyboard and vice versa. A direct comparison of the actions is not feasible due to their distinct nature so they cannot be used directly to measure usability. To make the comparison possible, we represent each action by a common component - time. This is the main advantage of our method, compared with a somewhat similar method described in [20]. In that paper, the physical effort is calculated as a sum of click counts, keyboard clicks and the number of pixels (at the mouse resolution) traversed by the user while moving the mouse from one point to another. The drawback of this method is that it is not possible to obtain a clear and unambiguous result while comparing programs. This happens because three different measures are being compared.

Our aim was to determine the average time required for each of the actions and then multiply the time by the recorded number of corresponding actions. Average action times were determined using a specialized application we developed, which gathers information of the time needed to perform an action from many users [19, 21]. At the time of writing, the information obtained from 2000 users is available.

We applied the described method to testing the efficiency of CAD systems. Contemporary CAD systems include a number of functions facilitating the users' needs. The manufacturers of this type of software usually emphasize the software capabilities, which leads to neglecting the ergonomic aspects of the software. This paper shows the results of this testing, which could be useful as guidelines for improvement of CAD systems.

The main contribution of the paper is the application of methodology for assessing the efficiency of the user interface of CAD systems and evaluation of these systems based on the presented methodology. Our method provides objectivity to usability assessment, while other methods are usually based on subjective user assessments. The existing comparisons of CAD systems are often based on the functions they include [22, 23]. In order to maintain objectivity, we limit the testing only to the efficiency of the user interface as it can be precisely measured and expressed in the common format, which is in this case the time required for action executions.

In contrast to the aforementioned subjective methods [13, 16], our method offers a numerical result of the analysis. Therefore, our approach makes it straightforward to compare user interfaces of different systems.

The application of our method is not limited to CAD systems. In the same manner, it can be applied to examine the efficiency of the user interface of any application. The objectivity of the assessment is the main advantage in these cases as well.

2. EFFICIENCY OF A CAD SYSTEM USER INTERFACE

Efficiency is commonly represented by the total time required to perform a specific task. Since the total time for performing a task depends also on user capabilities, knowledge, experience and familiarity, it is clearly a subjective measure. For instance, the user with a lot of experience with the system can perform the same task using shortcuts or an alternative function and thus complete the task faster.

The aforementioned efficiency refers to the efficiency of the full process of interaction with the software. In order to be as objective as possible, we only rely on determining the efficiency of the user interface, which we believe can be measured in an objective manner. The total time needed for process execution consists of the active time when the user communicates with the user interface of the software and passive time the user spends deciding the next step. The passive time depends on user's capabilities and his skills in the given software, but also on the software itself. The active time shows how good the user interface is for the given software. Using active time as a measure, we can in a more objective manner compare different programs and assess their user interface without analyzing the capabilities of the software itself.

Since the experiment analyzes interface efficiency of CAD systems, it is necessary to mathematically define efficiency in order to measure it. Efficiency can be defined in two ways: as a cost of design (Cd) divided by the effort (E) invested in the design, or as a relative efficiency (Er) which is calculated as a reciprocal of the effort. Since the cost of design is a variable category depending on numerous factors, while the relative efficiency can be used as a measure regardless of the type of project being analyzed, we opted for the latter.

Efficiency:

$$E_f = C_d/E \quad (1)$$

Specific efficiency:

$$E_r = 1/E \quad (2)$$

The effort occurring while using computer programs, including CAD systems, can be either mental or physical. Even though some methods for tracking mental effort also exist, measuring that kind of effort is necessarily subjective, which is a characteristic we tried to avoid. Therefore, we decided to measure only the physical effort which, in our opinion, can be objectively represented and measured.

The physical effort is related to the active communication between the user and the software. It can be represented as a sum of all actions executed by the user during that communication. If we were able to measure the time of those actions alone, we would obtain an objective measure of the efficiency of user interface.

A typical interface of computer programs is based on using a mouse and a keyboard as input devices. For communication with the computer program, users usually use one hand to choose between the mouse buttons or for scrolling, while the other hand is used to choose keys from the keyboard (ESC, SHIFT, ALT, CTRL). Moving the mouse is performed using the same hand used to choose the mouse buttons. If it becomes necessary for the user to enter textual data, the user will move their hand from the mouse and use both hands to type on the keyboard (see Fig. 1).

Based on the described usage routine, the physical effort during the usage of graphical user interface can be divided into several elementary actions. Those actions are mouse clicks, actions related to scrolling using the mouse, actions related to text entry from the keyboard, then actions related to moving the mouse cursor, as well as actions related to moving the hand between the mouse and the keyboard. When using CAD software, the actions frequently include holding the mouse button and moving the pointer at the same time. In this study, such actions were not considered, but the next version of the OpenClick software will include new tests which will be specific to the user interface of CAD systems.

The actions related to mouse clicks can be further split into left, right, middle button clicks, and double clicks. By analyzing the average time different users took to choose these buttons, we find that the average times differ for different button types, so these actions will be accounted for separately.

In actions related to scrolling the mouse, elementary effort originates from scrolling the mouse wheel for one step.

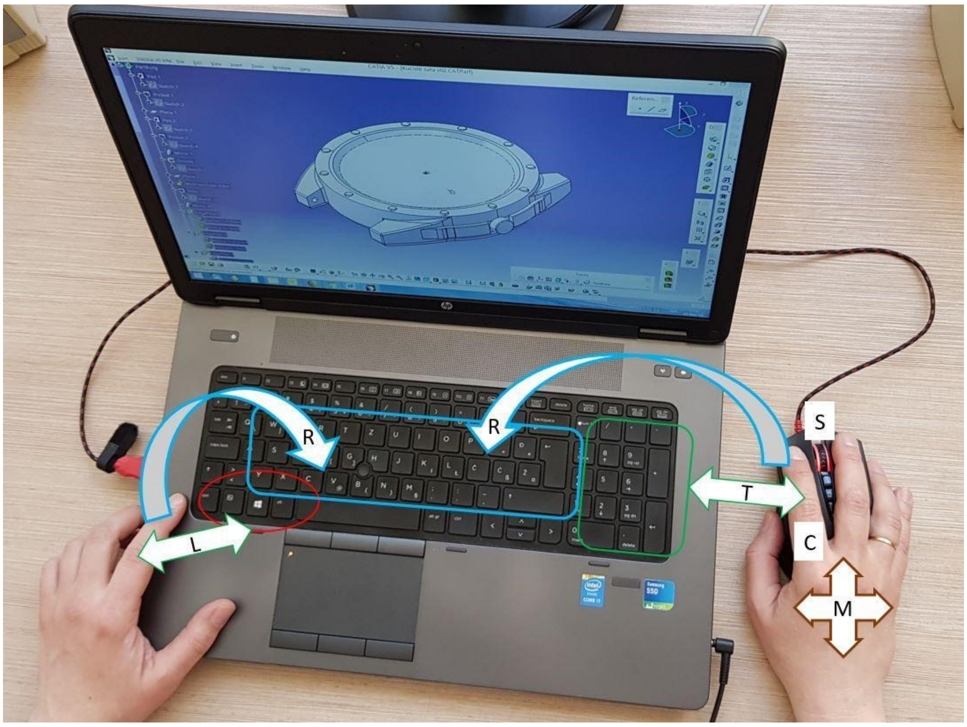


Fig. 1 Graphically presented control of input devices, where ‘C’ denotes mouse clicks, ‘S’ refers to scrolling, ‘L’ denotes choosing modifier keys, ‘R’ refers to text entry from the keyboard, ‘M’ to moving the hand with the mouse for positioning and ‘T’ refers to moving the hand between the mouse and the keyboard

Actions related to keyboard entries can be: (i) choosing one key, (ii) choosing multiple keys at once (as is the case for choosing keyboard shortcuts) and (iii) entering text or numbers.

Another elementary action we have measured is mouse movements. The goal of this action is to position the cursor on the desired location on the screen. The effort then comes from the need to move the hand with the mouse to the desired position.

The last type of action we investigated is related to switching between the mouse and the keyboard. This event occurs when the user needs to enter something using the keyboard after positioning the mouse, which is then followed by returning the hand to the mouse so that it can be positioned for the next action.

The total effort can be calculated as a sum of the efforts originating from individual actions of the user. The efforts from individual actions are determined as a sum of the time spent on the given action. Therefore, in order to determine the total effort, it is necessary to find the average time for individual actions. To that end, we have developed separate software described in more details in (Aranđelović at al. 2018b). The software was used to determine the average times of the following actions: left mouse click (tcl), right mouse click (tcr), middle mouse click (tcm), double click (tcd), mouse wheel scroll (ts), modifier key press (tla), choosing a combination of two keys from the keyboard (tlt), key press while typing (tk), moving the hand with the mouse to position the cursor (tm) and switching between the mouse and the keyboard (tt).

Using the number of individual elementary actions performed during the design process, as well as the average time needed to perform the corresponding action, the effort for each type of action can be defined.

As already mentioned, the total effort is directly proportional to the time spent on performing individual actions listed here and it is expressed in seconds:

$$E_{physical} \sim time_{active},\ t = t_{cl} + t_{cr} + t_{cm} + t_{cd} + t_{es} + t_{la} + t_{lt} + t_{k} + t_{m} + t_{t} \quad (3)$$

where tcl, tcr, tcm, tcd, tes, tla, tlt, tk, tm, tt are the times (efforts) per task corresponding to the average times defined above. Particular efforts can be computed using the formula (4).

$$E_{action} \sim \sum_{i=1}^{N_{action}} t_{action} \quad (4)$$
$$action \in \{cl, c, cm, cd, s, la, lt, k, m, t\}$$

N_{action} describes how many times the action was repeated.

2.1 Computing the average time of actions

As explained before, in order to compute the total effort, we need the average time required for performing the elementary actions. For that purpose, we developed a web application which includes various tests that correspond to elementary actions. As mentioned earlier, the actions that include holding the mouse button and moving the pointer were not directly analyzed in this application. For now, that effort is obtained by interpolating mouse clicks and movement, but in the new version of the software, we will add the tests covering these types of actions as well. This application has gathered data from over 2 000 users so far. They performed a total of over 30 000 tests. The characteristics of the users from whom the data were collected vary greatly, both in terms of age and gender, dominant hand (left or right), degree of computer knowledge, etc.

From 2001 users of the application, 1240 are male and 761 are female. There are 685 users younger than 30, 806 users between 30 and 50 and 510 users aged over 50.

The application includes tests which measure average time needed for mouse clicks (separately for left, right and middle key), average time required to choose a modifier key, average time needed to choose a combination of two keys, average time to enter text, etc. The application is available online so that it can attract more users (Mišić and Trajanović 2020).

Average action times are computed as average times from all users. The results for all users are stored in the database. The results of this analysis are shown in Table 1.

Table 1 Average duration of individual actions from the data collected with the OpenClick application

Test	Average value
Mouse speed	5417.448 (px/sec)
Left mouse button	0.975577 (s)
Right mouse button	0.940446 (s)
Middle mouse button	1.199912 (s)
Double click	1.422139 (s)
Shift key	1.141925 (s)
Esc key	1.141756 (s)
Alt key	1.162122 (s)
All modifier keys	1.154308 (s)
Combinations of keys	2.292802 (s)
Text entries	0.529892 (s)
Numeric entries	0.653146 (s)
Moving hand from mouse to keyboard	1.706697 (s)
Moving hand from keyboard to mouse	0.92824 (s)
Mouse scroll	0.11721 (s)

Before the analysis, the data were filtered manually to remove the entries that we believe are the consequence of irregular use of the software. The number of removed entries depends on the test type but is on average 4% for all tests.

3. EXPEPRIMENTAL

The proposed methodology was tested on four different CAD programs: Inventor, Solid Works, Fusion 360 and Creo. The choice of programs was based on the availability of software licenses and equally trained testers. The chosen group of programs may not be considered as fully representative concerning the CAD software market, as some of the leading programs (like CATIA or Siemens NX) were not tested. Nevertheless, as all the programs offer similar functions that are available through different user interfaces, and each of those hold a fair market share, we think that the group is representative.

All programs were installed on the same computer used for testing in order to remove a potential influence of hardware on the obtained results. Additionally, we used the latest versions of all systems to get a realistic comparison (Creo 6, Inventor 2020, SolidWorks 2019, Fusion 360 2.0.8809).

The efficiency of CAD programs depends not only on user interface, but also on the logic embedded in model building procedures. This typically leads to different steps in the creation of a sketch entity or a solid feature in various programs. Also, there are usually multiple procedures available for performing the same task in one program. As already explained, for the sake of objectivity, our task was to test only the efficiency of user interfaces. Thus, a number of simple test examples were created, where it was possible to perform all the basic steps in the creation of sketches and solid features in the same manner in all CAD programs. This meant that only those sketch elements that were available in all programs were used (e.g. rectangle defined by center and corner). This also meant that the chosen solid features could be created in each CAD software using the same basic modeling tools. The chosen program testers are experts in CAD and experienced in the software they were asked to use. They were also instructed to study and practice the modeling procedures before the final test runs were performed.

The testing procedure consisted of two sets of tests (one to evaluate sketches and another to evaluate features). The feature set of tests comprised 8 features and 3 sketches needed to create the features (Table 2), while the sketch set of tests comprised 9 tests, which were carried out inside the same sketch (Table 3).

A typical test example is shown in Fig. 2. In each CAD program, the three main steps in the creation of this simple 3D solid model, containing 2 solid features (the cube and the hole), were the same. In step I the centered rectangle sketch was created. In step II the sketch was extruded symmetrically in relation to the sketching plane, to obtain the cube. In step III, the hole with M8x1 ISO thread was created.

Table 2. Set of feature tests

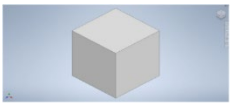
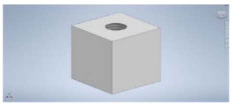
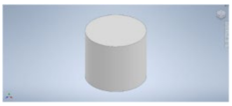
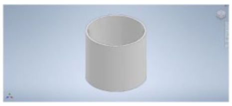
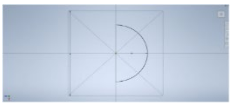
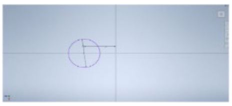
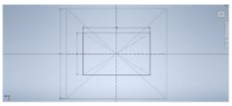
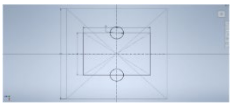
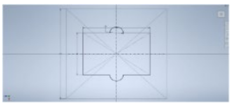
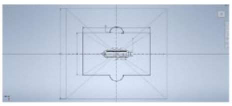
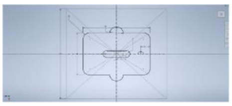
 Sketch (cube sketch)	 Extrude (cube)	 Hole (cube center threaded hole)
 Fillet (cylinder)	 Chamfer (cone)	 Shell (pipe profile)
 Draft (truncated cup)	 Sketch (sphere sketch)	 Sweep (torus)
 Sketch (pattern sketch)	 Extrude and pattern (pattern shape)	

Table 3. Set of sketch tests

 centerline	 rectangle	 Circle and mirror
 Trim and constraints	 Offset (from slot for which the process is not tracked)	 Fillet
 Construction geometry (inner slot)	 arc	 mirror

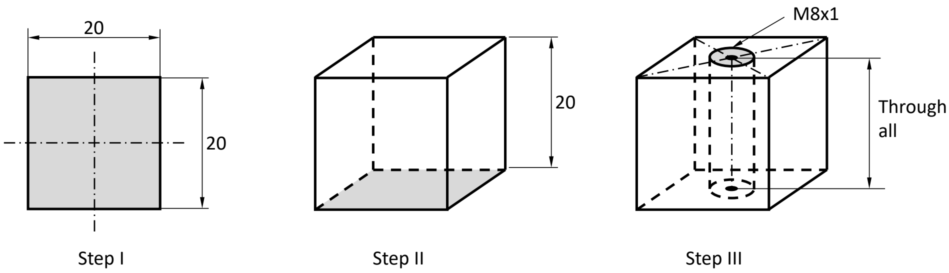


Fig. 2 Test example: creation of a centered cube with centered tapped hole

Each tester found the fastest way to complete each step of the example in the given CAD program, resulting in several elementary operations. As those operations differed in type and number between CAD programs, different times were calculated for each step depending on the software used. Some of the operations are identical in all programs, while some are not. Also, the (minimal) number of elementary operations varied from program to program.

The number of elementary actions and the path traversed were measured using the Mousotron program [24]. Each test was repeated 5 times and we used average values of the measured quantities.

4. RESULTS

In this section the results of the experiment are described. In Fig. 3, we show the total time needed to execute all tests. It can be seen that the least effort is required when using Inventor, while the most effort comes from the Creo software. As mentioned before, the efficiency is inversely proportional to the effort, so we conclude that Inventor is the software with the highest efficiency of the user interface. The results also show that CAD systems Inventor, Solid Works and Fusion 360 are relatively similar in the effort required, while only Creo deviates from this group.

The results show that there is a difference of around 20% between the efficiency of CAD systems. The detected difference can guide the manufacturers of the systems with weaker performance to improve specific aspects of their systems.

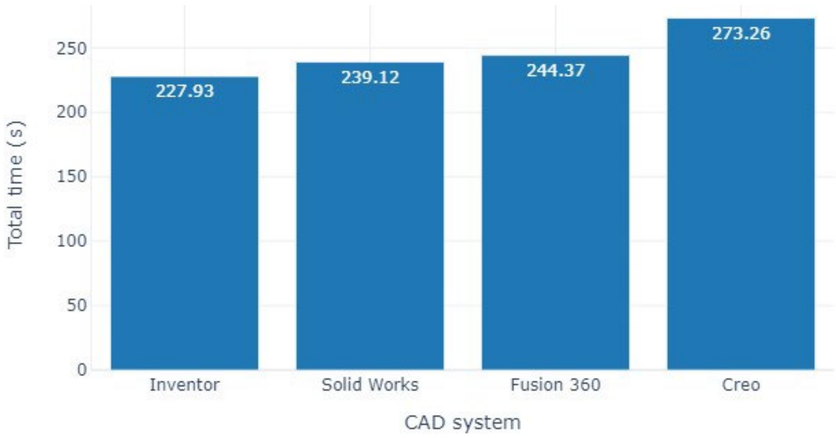


Fig. 3 Total time needed to execute all tests

Fig. 4 shows how the time was distributed between sketches and features for each program. The time shown represents the total time needed to design all sketches and features.

Fig. 4 Time for sketches and features

For producing sketches, Solid Works has the best performance, while for creating features, Inventor requires the least effort.

Fig. 5 shows the time spent on all elementary actions while Fig. 6 shows counts for individual actions.

Fig. 5 depicts how the time spent on interacting with the program is structured. The results show that most time is spent on actions related to mouse actions, followed by moving the hand between the mouse and the keyboard, keyboard-related actions, while the least amount of time is spent on positioning the mouse. It is important to note that the time spent on positioning the mouse is proportional to the path traversed by the mouse during the task. It is important to note that one of the contributing factors to the mouse positioning is the path traversed by the pointer while performing the task.

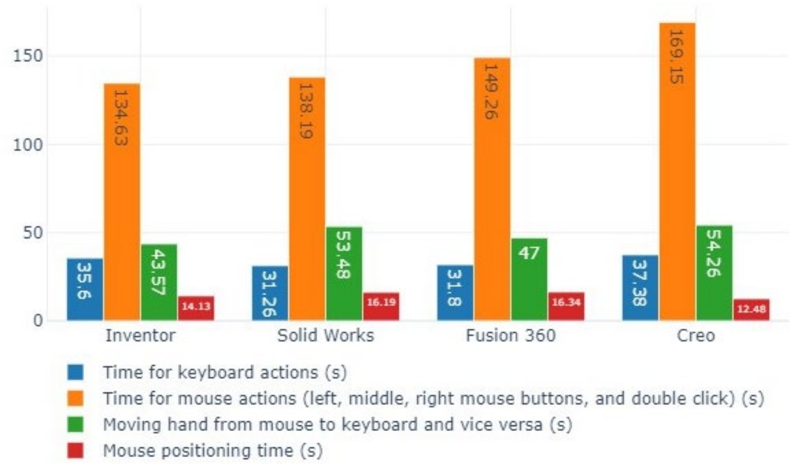


Fig. 5 Time by action types

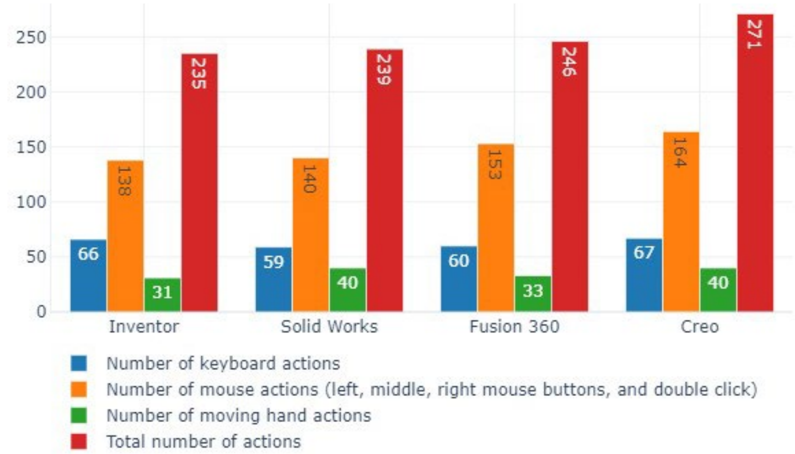


Fig. 6 Number of actions

Although in realistic conditions, engineers use the mouse wheel to zoom, we avoided this action in the test. Zooming was not used in order to prevent changes in tracked mouse movements during test repetitions (changes in displayed object or sketch size relative to the workspace size cause the mouse travel distance to oscillate for the same action).

Based on this, we can devise some guidelines for the improvement of the user interface. First, the number of mouse actions should be reduced and then the number of actions performed exclusively by using the mouse or the keyboard should be increased to avoid moving the hand between them.

Fig. 6 shows that most of the actions are related to pressing one of the mouse keys, followed by the actions related to the keyboard and then the actions of moving the hand from the mouse to the keyboard and vice versa. These results were obtained by counting elementary actions, while the pointer movement simultaneously with the mouse click was not considered.

As shown in Fig. 5 and Fig. 6, even though there are almost twice as many actions related to the keyboard entries as the actions of moving the hand between the keyboard and the mouse, the time spent on moving the hand is still longer. That happens due to the longer average time required for moving the hand.

The count and duration of actions were analyzed on creating simple elements and sketches. As shown earlier, even in these cases, the user interface of some systems is more efficient than those of other systems. In more realistic cases, when complex elements and assemblies are created, the difference in efficiency becomes notable. These elements are obtained as a combination of sketches and features we used in this experiment. If there is a difference in efficiency while creating simple sketches and features, then that difference will definitely be larger once it is necessary to combine multiple sketches and features present in assemblies.

Another interesting aspect of this analysis is the treatment of errors made while creating the elements. For the experiment described in this work, we defined the exact number of actions needed to produce each sketch or feature in advance. That was possible because the procedures were precisely defined so that they were possible to follow in all tested systems, and such procedures could be defined because all elements were simple. Our test environment could be considered ideal since the users' errors could be safely discarded. In more realistic conditions, the users will occasionally make errors which will influence the active time spent in communication with the user interface. Although the number of errors is subjective and depends on the user, the probability of the error increases with the number of actions needed to complete the task. Therefore, we conclude that for CAD systems where the number of actions is different, the difference in the efficiency of user interface will be larger in realistic conditions, due to the increased number of errors.

Fig. 7 shows the percentage of time required to perform individual actions with respect to the total time. The percentages are similar for all programs, so it can be concluded that approximately 60% of time is spent on actions related to the mouse, 13-15% of the time are spent on keyboard actions, 20% on moving the hand between the mouse and the keyboard and approximately 6% on positioning the mouse.

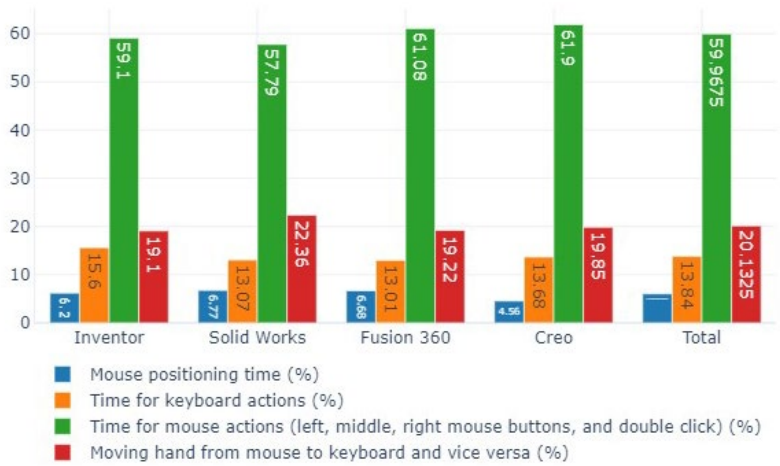


Fig. 7 The percentage of time required to perform individual actions with respect to the total time

5. CONCLUSION

To examine the efficiency of CAD systems, we employed our time-based methodology, based on determining the number of elementary actions during the communication between the user and the CAD program for a predefined set of tasks. For all elementary actions, we experimentally determined the average times using a separate application we developed. By multiplying the number of actions and the average action times we devised the amount of active time while using different CAD programs.

The method was tested on 4 different CAD systems used at our faculty. The tasks in the CAD systems were performed by experts with extensive experience with the corresponding software. For testing purposes, we designed an experiment including a large number of sketches and features in all of these CAD systems. We chose sketches and features which are most frequently used. For all these tasks we defined the same procedure, as our aim was not to test the capabilities of the programs, but to test their user interface.

The conducted experiment showed that the developed methodology could be used to perform an objective comparison of the efficiency of user interface. In this work, the proposed method was applied to assess the interface of CAD systems, but it can be equally applied to other programs. The method introduces objectivity to the analysis which is generally very subjective.

In the future work, we plan to include more CAD systems in the evaluation and extend the result base. Additionally, we will introduce more complex elements to the testing procedure, in order to measure the workflow in a more realistic manner. The challenge with this type of tests is to define a common procedure which would enable objective assessment. Designing complex assemblies and elements in any CAD system can be done in multiple different ways, which requires attention and experience when planning the experiment.

5. References

1. ISO 9241-11. (2018). Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts. International Organization for Standardization. Geneva
2. Usability evaluation. (2020). Usability evaluation methods. Retrieved October 20, 2020. https://www.usabilityhome.com/
3. Hartson, R., Pardha, P., 2018, *UX Evaluation methods and techniques,* In The UX Book: Agile UX Design for a Quality User Experience (2nd ed.), Morgan Kaufmann
4. Budiu, R., 2017, *Quantitative vs. Qualitative Usability Testing,* Nielsen Norman Group. https://www.nngroup.com/articles/quant-vs-qual
5. Hornbæk, K., 2006, *Current practice in measuring usability: Challenges to usability studies and research,* International Journal of Human Computer Studies, 64, pp. 79-102
6. Nielsen, J., 2001, *Usability metrics,* Nielsen Norman Group. https://www.nngroup.com/articles/usability-metrics
7. Card, S., Moran, T., Newell, A., 1980, *The keystroke-level model for user performance time with interactive systems,* Communications of the ACM, 23(7), pp. 399. https://doi/org/10.1145/358886.358895
8. Kieras, D., 2001, *Using the Keystroke-Level Model to Estimate Execution Times,* (Report) https://www.researchgate.net/publication/2848715 Using the Keystroke Level Model to Estimate Execution Times
9. Allison, R., Hayes, C., McNulty, C., Young, V., 2019, *A Comprehensive Framework to Evaluate Websites: Literature Review and Development of GoodWeb,* JMIR Formative Research 3(4), https://doi.org/10.2196/14372
10. Weichbroth, P., 2020, *Usability of mobile applications: A systematic literature study,* IEEE Access, 8, 55563-55577
11. Li, Y., Hedlind, M., Kjellberg, T., 2015, *Usability evaluation of CADCAM: State of the art,* Proceedings of the CIRP 25th Design Conference Innovative Product Creation, pp. 205-210.
12. Chilana, P., Wobbrock, J., Ko, A., 2010, *Understanding Usability Practices in Complex Domains,* In CHI 2010: Usability Methods and New Domains, pp. 2337-2446
13. Luczak, H., Springer, J., Shmidt, L., 2006, *Ergonomics of CAD systems,* In W. Karwowski (ed.), International Encyclopedia of Ergonomics and Human Factors (pp. 1088-1093), Taylor & Francis
14. Ulman, D., 2002, *Toward the ideal mechanical engineering design support system,* Research in Engineering Design, 13, pp. 55-64.
15. Sung, R., Ritchie, J., Rea, H., Corney, J., 2011, *Automated design knowledge capture and representation in single-user CAD environments,* Journal of Engineering Design, 22, pp. 487–503.
16. Lee, G., Eastman, C., Taunk, T., Ho, C., 2010, *Usability principles and best practices for the user interface design of complex 3D architectural design and engineering tools,* International Journal of Human Computer Studies, 68, pp. 90-104.

17. Attia, S., Andersen, M., 2013, *Measuring the Usability, Efficiency and Effectiveness of CAAD Tools and Applications,* In ECAADe, 31st International Conference on Education and Research in Computer Aided Architectural Design in Europe
18. Aranđelović, J., Drašković, P., Turudija, R., Dimitrov, M., Božić, N., Korunović, N., Mišić, D., Trajanović, M., 2018a, *Trial experimental determination oft he average times of actions executed in a CAD application,* In 37th International Conference on Production Engineering, Serbia.
19. Aranđelović, J., Drašković, P., Turudija, R., Dimitrov, M., Božić, N., Korunović, N., Trajanović, M., 2018b, *Towards a Methodology for CAD program Efficiency Assessment,* Proceedings of 4th International Conference Mechanical Engineering in XXI Century, pp. 155-161, Serbia.
20. Komogortsev, O., Mueller, C., Tamir, D., Feldman, L., 2009, *An effort based model of software usability,* In International Conference on Software Engineering Theory and Practice (SETP-08).
21. Misić, D., Trajanović, M., 2020, *OpenClick,* Faculty of Mechanical Engineering Nis. Retrieved December 1, 2020. http://openclick.masfak.ni.ac.rs
22. Aermech, 2015, *Which software is better CATIA vs CREO vs SOLIDWORKS?,* https://aermech.com/which-software-is-better-catia-vs-creo-vs-solidworks-aermech-comparison
23. Wikipedia, 2020, *Comparison of 3D computer graphics software,* Retrieved October 20, 2020, https://en.wikipedia.org/wiki/Comparison_of_3D_computer_graphics_software
24. Blacksun Software, 2020, *Mousotron,* (version 12.1) [Computer Software] http://www.blacksunsoftware.com/mousotron.html