

# Teaching Computerized Score Editing Using the Keystroke-Level Model / Predarea tehnoredactării muzicale utilizând modelul Keystroke-Level

Traian PENCIUC

Music Department, Faculty of the Arts in Romanian Language, University of the Arts in Târgu Mureș / Departamentul de Muzică,  
Facultatea de Arte în Limba Română, Universitatea de Arte din Târgu Mureș  
penciuctraian@gmail.com

## ABSTRACT

*This paper presents a method for incorporating the Keystroke-Level Model into the didactic approach by using algorithmization in teaching computerized score editing, specifically within the Sibelius application. The paper discusses how the operator categories of the Keystroke-Level Model can enhance the design of instructional algorithms and help students become more aware of the actions required in the complex and sensitive interface of these applications. The study emphasizes the importance of the M (mental preparation) and R (system response) operators, which can be adapted to create moments when the students understand the application's response and coordinate their actions and decisions. Lastly, the paper explores the role of the Keystroke-Level Model in modeling algorithms suitable for student training and quasi-algorithms used by advanced users.*

## Keywords

Human Computer Interaction, Keystroke-Level Model, computer score edit teaching, algorithmization, Sibelius.

## INTRODUCTION

*Human-Computer Interaction (HCI) is a field that focuses on interfaces between humans and computers. Starting from the observation that computer systems are tools with high complexity, it seeks to understand aspects of this interaction: interface design, defining commands and the sequence in which they are applied, analysing the flow of information between parties, and user actions, but also representations they have on the operation of the software. These studies aim to reduce usage errors and to obtain ergonomic and functional optimization.*

*This paper will explore the didactic utility of HCI analysis in teaching score editing, focusing on the simplest HCI model, namely the Keystroke-Level Model (KLM). As its name suggests, the method breaks down the user's actions into primitive gestures, which makes the student aware of each stage of executing a command. It is also helpful to the instructor in describing the algorithms of the commands and designing them for learning these commands, contributing to their clarity, and achieving an optimal balance between completeness and simplicity. It can also answer how learning*

## REZUMAT

*Această lucrare prezintă o metodă de incorporare a modelului Keystroke-Level (KLM) în abordarea didactică prin utilizarea algoritimizării în predarea tehnoredactării muzicale computerizate, în special în cadrul aplicației Sibelius. Lucrarea discută modul în care categoriile de operatori ale modelului Keystroke-Level pot îmbunătăți proiectarea algoritmilor de instruire și pot ajuta studenții să devină mai conștienți de acțiunile necesare în relație cu interfața complexă și sensibilă a acestor aplicații. Studiul subliniază importanța operatorilor M (pregătirea mentală) și R (răspunsul sistemului), care pot fi adaptați pentru a crea momente în care studenții înțeleg răspunsul aplicației și își coordonează acțiunile. În cele din urmă, lucrarea explorează rolul modelului Keystroke-Level în modelarea algoritmilor adecvați pentru formarea studenților și a cvasi-algoritmilor utilizați de utilizatorii avansați.*

## Cuvinte cheie

Interacțiunea om-calculator, modelul Keystroke-Level, tehnoredactare muzicală computerizată, algoritimizare, Sibelius.

## INTRODUCERE

*Interacțiunea om-calculator (Human-Computer Interaction – HCI) este un domeniu care se concentrează pe interfețele dintre oameni (utilizatori) și computere. Pornind de la observația că sistemele informatice sunt unelte cu o complexitate ridicată, se caută să se înțeleagă aspecte privind această interacțiune: designul interfețelor, definirea comenzilor și succesiunea în care sunt aplicate, analiza fluxului de informații între părți, a acțiunilor utilizatorului, dar și a reprezentărilor pe care acesta le are asupra funcționării softurilor. Țelul acestor studii este diminuarea erorilor de utilizare, optimizarea ergonomică și funcțională.*

*În această lucrare vom explora utilitatea didactică a analizei HCI în predarea tehnoredactării partiturilor muzicale, concentrându-ne asupra celui mai simplu model HCI și anume modelul [analizei] la nivel de tastare (Keystroke-Level Model – KLM). Așa cum sugerează și denumirea ei, metoda descompune acțiunile utilizatorului în gesturi singulare, ceea ce face ca studentul să conștientizeze fiecare etapă a execuției unei comenzi. Ea este utilă și instructorului, la descrierea algoritmilor comenzilor și la realizarea designului algoritmilor de învățare a acestor comenzi, contribuind la claritatea acestora, la atingerea unui nivel optim între completitudine și simplitate. Aceasta poate totodată răspunde felului în care algoritmii de învățare pot fi*

algorithms can be adapted according to the student's level of training and computer skills.

Although the method presented in this paper can be extended to other applications, we will adapt it concretely for the *Sibelius* score editor. This paper's examples and interface descriptions refer to version 7.5 of this application.

## THE KEYSTROKE-LEVEL MODEL

The *Keystroke-Level Model* (KLM) was theorized by Embley, Lan, Leinbaugh, and Nagy (1978), and by Card, Moran, and Newell (1980, 1980a, and 1983). This model analyses the application at the level of primitive commands – typing or mouse clicks, each task being broken down into commands and commands into mouse keystrokes or actions (Embley *et al.*, 641).

KLM's original purpose was as a tool for estimating lead times by users together with IT systems. The researchers studied the sub-tasks and the physical-motor operators in detail, including the system's response time and, heuristically, the time required to mentally prepare the user for the operations.

Finally, they defined six categories of actions, called *operators*, which are primitive actions of any task (Card, Moran, and Newell, 1983, 261-267):

- K – Keystroke or button press: pressing keys on the numeric or alphanumeric keypad, alternate buttons (Control, Shift, Alt), or combinations thereof. The duration of a type depends on the user's skill level;
- P – pointing the mouse in a particular position or on a specific item on the screen;
- H (*homing*) – (re)placing your hand on the keyboard, mouse or other device, regardless of the direction of travel. Later, Burns, Ritter, and Zhang (2016) would differentiate between placing one's hand on the keyboard versus placing one's hand on the mouse;
- D (*drawing*) – mouse movement on a specific trajectory;
- M – mental preparation for the execution of a physical action. It refers to thinking, remembering specific parameters or making decisions about the following operation;
- R – application response.

KLM is used to optimize software interfaces by making predictions about the time it takes for the user to complete a task interactively with the app. To this end, the researchers assigned each operator a standard duration (Card, Moran, and Newell, 1983, 264) based on heuristic deductions that did not suffer dramatic results in subsequent work (see. Al-Megren, Khabati and Al-Khalifa, 2018). To estimate a task's completion time, it is decomposed into primitives from the six categories of operators, the execution time being calculated as the sum of the standard duration of each primitive action.

Next, we will use KLM to compare two methods of inputting simple musical notes in *Sibelius*. The first uses the mouse: the user selects the note value with a click on the corresponding button in the *Keypad pane* and then places it on the keyboard with another click in the desired place. The second uses the keyboard exclusively. First, the user selects the note value by pressing the

adaptați în funcție de nivelul de pregătire și de îndemânarea informatică a studentului.

Deși metoda prezentată în această lucrare poate fi extinsă și asupra altor aplicații, noi o vom adapta concret pentru editorul de partituri *Sibelius*. Exemplele și descrierile interfeței din această lucrare se referă la versiunea 7.5 a acestei aplicații.

## MODELUL LA NIVEL DE TASTARE (KEYSTROKE-LEVEL MODEL)

Modelul [analizei] la nivel de tastare (*Keystroke-Level Model* – KLM) fost teoretizat de Embley, Lan, Leinbaugh și Nagy (1978) și de Card, Moran și Newell, (1980, 1980a și 1983). Acest model analizează aplicația la nivelul comenzilor primare – tastări sau clicuri de mouse, fiecare sarcină fiind descompusă în comenzi, iar comenzile, în tastări sau acțiuni ale mouse-ului (Embley *et al.*, 641).

Destinația inițială a KLM a fost aceea de instrument de estimare a timpilor de execuție de către utilizatori împreună cu sistemele informatice. Cercetătorii au studiat în amănunt sub-sarcinile, operatorii fizico-motorii, incluzând și timpul de răspuns al sistemului și, euristic, timpul necesar pregătirii mentale a utilizatorului pentru operație.

În sfârșit, au definit șase categorii de acțiuni, numite *operatori*, care sunt acțiuni primitive ale oricărei sarcini (Card, Moran și Newell, 1983, 261-267):

- K – Apăsări de taste sau butoane (*keystroke* sau *button press*): apăsarea unor taste din tastatura numerică sau alfanumerică, a butoanelor alternative (Control, Shift, Alt) sau a combinațiilor acestora. Durata unei tastări depinde de nivelul de îndemânare al utilizatorului;
- P – așezarea mouse-ului (*pointing*) într-o anumită poziție sau pe un anumit item de pe ecran;
- H (*homing*) – (re)așezarea mâinii pe tastatură, mouse sau alt obiect, indiferent de direcția de deplasare. Mai târziu Burns, Ritter și Zhang (2016) vor diferenția așezarea mâinii pe tastatură față de așezarea mâinii pe mouse;
- D (*drawing*) – mișcarea mouse-ului pe o anumită traiectorie;
- M – pregătirea mentală pentru executarea unei acțiuni fizice. Se referă la gândire, memorarea unor parametri sau luarea unor decizii referitoare la operația care urmează;
- R – răspunsul aplicației.

KLM este folosită în optimizarea interfețelor softurilor prin predicții privind timpul necesar de utilizator ca, interactiv, împreună cu aplicația, să ducă la bun sfârșit o sarcină. În acest scop, cercetătorii au atribuit fiecărui operator o durată standard (Card, Moran și Newell, 1983, 264) în urma unor deducții euristice, care nu au suferit modificări consistente în lucrările ulterioare (Al-Megren, Khabati și Al-Khalifa, 2018). Pentru a estima timpul îndeplinirii unei sarcini, aceasta este descompusă în acțiuni primitive din cele șase categorii de operatori, timpul de execuție fiind calculat ca suma duratei standard a fiecărei acțiuni.

În continuare vom utiliza KLM pentru a compara două metode de introducere a notelor muzicale simple în aplicația *Sibelius*. Prima utilizează mouse-ul: utilizatorul selectează valoarea notei cu un clic pe butonul corespunzător din panoul *Keypad* și apoi o așază pe portativ cu un alt clic în locul dorit. Cea de a doua utilizează exclusiv tastatura. Mai întâi utilizatorul

corresponding key in the numeric keypad, which he/she intuitively because the location of the buttons in the *Keypad* pane mirrors the numeric keypad. Then, he/she writes the note by typing the letter corresponding to the note in the Anglo-Saxon system.

The novice user prefers the first input method due to familiarity with the mouse and not remembering successions of keystrokes. Moreover, the Romanian musician, educated in the solfege system, will have difficulties in alphabetically encoding the notes's pitch. However, textbooks for writing scores (Clark and Spreadbury, 2009; *Sibelius 7.5 Reference Guide*; Penciu, 2019) recommend good practice writing from the keyboard, emphasizing without facts that, over time, keyboard writing speed becomes superior to mouse use. Through the KLM analysis, we will compare the effectiveness of the two input methods.

The placement of notes with the mouse breaks down into the following operations considered for copying a note or musical sign (reaction times R are neglected):

**Table 1. KLM analysis for a musical note input using the mouse**

Cod	Operation	Standard duration (s) <sup>1</sup>
M <sub>1</sub>	Read the pitch and duration of the note in the manuscript	1.35s
P <sub>1</sub>	Click the duration icon in the <i>Keypad</i> pane	1,1s
P <sub>2</sub>	Click on the position where the note should be placed	1,1s
Total duration M <sub>1</sub> +P <sub>1</sub> +P <sub>2</sub>		3.55s

For the novice user, unfamiliar with the *Sibelius* keyboard, the analysis looks like this:

**Table 2. KLM analysis for a musical note input using the keyboard by a novice user**

Cod	Operation	Standard duration (s)
M <sub>1</sub>	Read the pitch and duration of the note in the manuscript	1.35s
M <sub>2</sub>	Remember or deduce the key in the numeric keypad that must be pressed to select the note duration	1.35s
K <sub>1</sub>	Press that key	0.28s
M <sub>3</sub>	Remember the key that must be pressed to select the pitch of the note	1.35s
K <sub>2</sub>	Press that key	0.28s
Total duration M <sub>1</sub> +M <sub>2</sub> +K <sub>1</sub> +M <sub>3</sub> +K <sub>2</sub>		4.61s

This analysis shows another reason why entering notes from the keyboard is unfamiliar to beginners. The number of operations is higher and involves a higher psychic effort (M<sub>1</sub> and M<sub>2</sub>). Moreover, the standard writing speed (4.61s / sign – 13 signs / min.) is lower than that in the mouse input version (3.55s / sign – 16 signs / min.).

<sup>1</sup> Standard durations are set by Card, Moran, and Newell (1980a) and Card, Moran, and Newell (1983, 264).

selectează valoarea notei apăsând tasta corespunzătoare din tastatura numerică, pe care o intuiește deoarece amplasarea butoanelor din panoul *Keypad* oglindește tastatura numerică. Apoi scrie nota tastând litera corespunzătoare notei în sistemul anglosaxon.

Utilizatorul începător o preferă pe prima, deoarece este obișnuit să utilizeze mouse-ul, nu să rețină succesiuni de apăsări de taste. Mai mult, muzicianul român, educat în sistemul *solfege*, va avea dificultăți în codificarea alfabetică a înălțimii notelor. Totuși, manualele de scriere de partituri (Clarke și Spreadbury, 2009; *Sibelius 7.5 Reference Guide*; Penciu, 2019) recomandă ca o practică bună scrierea din tastatură, subliniind, fără a argumenta, faptul că, în timp, viteza de redactare din tastatură devine superioară utilizării mouse-ului. Prin analiza KLM vom compara eficiența celor două metode de introducere.

Așezarea notelor cu mouse-ul se descompune în următoarele operații considerate pentru copierea unei note sau semn muzical (timpii de reacție R sunt neglijați):

**Tabel 1. Analiza KLM pentru metoda introducerii unei note muzicale cu ajutorul mouse-ului**

Cod	Operație	Durată standard (s)
M <sub>1</sub>	Citește înălțimea și durata notei în manuscris	1,35s
P <sub>1</sub>	Clic pe iconița corespunzătoare duratei din panoul <i>Keypad</i>	1,1s
P <sub>2</sub>	Clic în poziția în care se dorește așezarea notei	1,1s
Durată totală M <sub>1</sub> +P <sub>1</sub> +P <sub>2</sub>		3,55s

Pentru utilizatorul începător, nefamiliarizat cu tastatura *Sibelius*, analiza arată astfel:

**Tabel 2. Analiza KLM pentru metoda introducerii unei note muzicale cu ajutorul tastaturii de către un utilizator începător**

Cod	Operație	Durată standard (s) <sup>1</sup>
M <sub>1</sub>	Observă înălțimea și durata notei în manuscris	1,35s
M <sub>2</sub>	Își amintește sau deduce tasta care trebuie apăsată pentru a selecta durata notei	1,35s
K <sub>1</sub>	Apasă tasta respectivă	0,28s
M <sub>3</sub>	Își amintește tasta alfanumerică care trebuie apăsată pentru a selecta înălțimea notei	1,35s
K <sub>2</sub>	Apasă tasta respectivă	0,28s
Durată totală M <sub>1</sub> +M <sub>2</sub> +K <sub>1</sub> +M <sub>3</sub> +K <sub>2</sub>		4,61s

Observăm din această analiză un alt motiv pentru care introducerea notelor din tastatură este nefamiliară începătorilor. Numărul de operații este mai mare, implică un efort psihic (M<sub>1</sub> și M<sub>2</sub>) mai mare, iar viteza standard de scriere (4,61s / semn – 13 semne / min.) este mai mică decât cea în varianta utilizării mouse-ului (3,55s / semn – 16 semne / min.).

<sup>1</sup> Timpii standard sunt stabiliți de Card, Moran și Newell, (1980a, 1983).

However, while the speed of writing with the mouse remains constant, as the student practices writing from the keyboard, their writing speed will increase significantly. This is due to the automatisms that are installed through practice. Like a pianist, the student will press the necessary keys without being aware of the gesture. Thus, operations  $M_2$  and  $M_3$ , in which they decide which key to press, disappear, while  $M_1$ , in which the note is prepared, does not; thus, only the writing of the note will remain. The total duration under these conditions becomes

$$T = M_1 + K_1 + K_2 = 1,51s / \text{sign}$$

and corresponds to a considerably improved writing speed of 39 signs/min.

If we continue the analysis at a more advanced level, the expert user will be able to memorize a group of notes, thus rarely turning their gaze to the manuscript or even being able to write without looking at the keyboard, making  $M_1$  operative time negligible. According to Card, Moran, and Newell (1983, 264), the one-key typing time reduces to 0.08s. In these circumstances, at the limit, the total duration is

$$T = K_1 + K_2 = 0,16s / \text{sign}$$

corresponding to an excellent writing speed of 375 signs / min.

These values are not guaranteed; they depend on the user's motor skills and experience. The practice has proven that as they write more, the speed increases, and after periods of interruption, it decreases. However, as in practice, the KLM model demonstrates the superiority of keyboard over mouse input in entering notes.

## KLM AND ALGORITHMIZATION

The use of KLM analysis in software teaching is related to the algorithmization method in that both break down the task into successive actions. This closes a circle, algorithmization being a teaching and learning method borrowed from computer science.

The algorithm is defined as a "generally comprehensible prescription" composed of a "sequence of elementary operations" in order to achieve a goal (Landa, 1974, 11).

Algorithmization is the transposition in the didactic process of the formal algorithm, taking from it the stepwise, logical, and chronological sequence principle to achieve a result.

Algorithmization is the most straightforward and most pragmatic method of achieving a result. It requires the user to perform a well-determined sequence of operations to achieve the result. The algorithm does not require knowledge about the application's operation; it can be applied mechanically. That is why algorithmization is an excellent method for beginners and advanced learners who want to get quick results.

In Kopstein's simple definition, "algorithmization provides the methodology – the detailed *how-to* – for instructional design" (Kopstein, 1974, 13).

In the didactics of using software, algorithmization is based on representing it as an automaton that executes commands. From this perspective, the algorithm is just a recipe that brings a result.

Însă, pe când viteza de scriere cu mouse-ul rămâne aproximativ constantă, pe măsură ce studentul practică scrierea din tastatură, viteza de redactare a acestuia va crește simțitor. Acest fapt se datorează automatismelor ce se instalează prin exersare. La fel ca un pianist, studentul va apăsa tastele necesare fără să fie nevoit să conștientizeze gestul. Dispar astfel operațiile  $M_2$  și  $M_3$ , în care acesta decide ce tastă apasă, în afară de  $M_1$ , operațiunea prin care este pregătită nota; va rămâne astfel doar scrierea notei. Durata totală în aceste condiții devine

$$T = M_1 + K_1 + K_2 = 1,51s / \text{semn},$$

care corespunde unei viteze de scriere de 39 semne / min., considerabil îmbunătățită.

Dacă continuăm analiza, utilizatorul expert va fi capabil să memoreze un grup de note, ducând astfel mai rar privirea către manuscris, sau chiar e capabil să scrie fără să privească tastatura. Prin urmare, timpul operației  $M_1$  devine neglijabil. După Card, Moran și Newell (1983, 264), timpul de tastare cu o singură tastă se reduce la 0,08s. În aceste condiții, la limită, durata totală este

$$T = K_1 + K_2 = 0,16s / \text{semn},$$

ceea ce corespunde unei viteze de scriere ridicate, de 375 semne / min.

Aceste valori nu sunt garantate, ele depinde de aptitudinile motorii ale utilizatorului și de experiența acumulată. Practica a dovedit că, pe măsură ce scrie mai mult, viteza crește, iar după perioade de întrerupere aceasta scade. Totuși, la fel ca în practică, și modelul KLM demonstrează superioritatea utilizării tastaturii față de mouse în introducerea notelor.

## KLM ȘI ALGORITMIZAREA

Utilizarea analizei KLM în pedagogia utilizării softurilor este legată de metoda algoritmicizării, prin faptul că ambele descompun sarcina în acțiuni succesive. Se încheie astfel un cerc, algoritmicizarea fiind o metodă de predare și învățare împrumutată din informatică.

Algoritmul este definit ca o „rețetă precisă și inteligibilă”, compusă dintr-o „succesiune de operații elementare” în scopul de a atinge un obiectiv (Landa, 1974, 11).

Algoritmicizarea este transpunerea în procesul didactic a algoritmului formal, preluând de la acesta principiul treptat, succesiunea coerentă, logică și cronologică, cu scopul de a ajunge la un rezultat.

Algoritmicizarea este cea mai simplă și pragmatică metodă pentru a îndeplini o sarcină. Este suficient ca utilizatorul să efectueze o succesiune bine determinată de operații pentru a atinge rezultatul. Algoritmul nu cere cunoștințe deosebite despre felul în care funcționează aplicația, acțiunile putând fi realizate mecanic. De aceea, algoritmicizarea este o metodă foarte bună pentru începători, dar și pentru avansați, atunci când aceștia din urmă vor să obțină rezultate rapide.

În definiția simplă a lui Kopstein „algoritmicizarea furnizează metoda – acel *cum se face*, descris în detaliu – pentru proiectarea instruirii” (Kopstein, 1974, 13).

În didactica utilizării unui soft, algoritmicizarea se bazează pe execuția automată a comenzilor. Din această perspectivă, algoritmul este doar o rețetă care aduce un rezultat.

Directly applicable to applications with simple interfaces, the algorithm requires additions in complex applications to ensure their comprehensibility to a level where the user can master them creatively, not automatically. For beginners, additions can refer to aspects that prevent confusion or mishandling that causes errors. This results in the instructional algorithm that the teacher presents to the student so that they can acquire the working algorithm.

KLM can be helpful in developing learning algorithms. In this approach, we are not interested in time calculations, but in their preliminary stage, the decomposition of tasks into actions, for which KLM offers efficient tools that increase the resolution of this fragmentation to primitive operations that may be relevant in the instructional approach. Moreover, by introducing the mental activities of the user operator M and the operator R that describes the reaction of the application, the psychic actions of the user (observation, decision, preparation) are distinctly pointed out, which are added to the physical ones (typing, pointing, clicking). Here are some factors by which KLM can improve the design and learning of instructional algorithms.

The precision with which the analysis sequences operations ensures the student's awareness of each action, and the teacher provides a framework that eliminates gaps in the development of learning algorithms that can be implanted in the case of applications with complex interfaces.

Such an approach also applies to the development of instructional algorithms for *Sibelius*. I have written about the application's complexity before, emphasising the need for specific didactic approach methods but also for the continuous updating of best practices (Penciu, 2019). Appreciations in this regard can also be found in Clark and Spreadbury (passim), and Metea (2010) summarizes the various possibilities of the application. I will mention here just a few of these traits that require more careful algorithmization:

- The diversity of scores and musical genres they cover;
- The existence of hidden or dissimulated controls in properties menus;
- The interface elements serving the same tasks are not always presented as grouped;
- The effect of the most common keys can be unexpected. Unlike a text editor, even keys corresponding to alphabet letters generate commands. For example, the R key repeats a written passage, S initiates a slur, N activates the notation mode, etc. Then, the digit keys in the alphanumeric keypad activate commands other than those in the numeric keypad. For this reason, the interface is susceptible to erroneous keystrokes that have undesirable effects that are, very likely, also unknown to the user, given the complexity of the application. The fact that from the first courses, the student learns that they can cancel the effect of an inadvertently pressed key with Undo (Ctrl-Z) or Escape gives them security and a sense of control of the application that encourages learning. However, the student must form an

Applicable direct in the case of applications with simple interfaces, the algorithm requires additions in complex applications to ensure their comprehensibility to a level where the user can master them creatively, not automatically. For beginners, additions can refer to aspects that prevent confusion or mishandling that causes errors. This results in the instructional algorithm that the teacher presents to the student so that they can acquire the working algorithm.

KLM can be helpful in developing learning algorithms. In this approach, we are not interested in time calculations, but in their preliminary stage, the decomposition of tasks into actions, for which KLM offers efficient tools that increase the resolution of this fragmentation to primitive operations that may be relevant in the instructional approach. Moreover, by introducing the mental activities of the user operator M and the operator R that describes the reaction of the application, the psychic actions of the user (observation, decision, preparation) are distinctly pointed out, which are added to the physical ones (typing, pointing, clicking). Here are some factors by which KLM can improve the design and learning of instructional algorithms.

The precision with which the analysis sequences operations ensures the student's awareness of each action, and the teacher provides a framework that eliminates gaps in the development of learning algorithms that can be implanted in the case of applications with complex interfaces.

Such an approach also applies to the development of instructional algorithms for *Sibelius*. I have written about the application's complexity before, emphasising the need for specific didactic approach methods but also for the continuous updating of best practices (Penciu, 2019). Appreciations in this regard can also be found in Clark and Spreadbury (passim), and Metea (2010) summarizes the various possibilities of the application. I will mention here just a few of these traits that require more careful algorithmization:

- Diversitatea partiturilor și a genurilor muzicale pe care le acoperă;
- Existența unor comenzi ascunse sau disimulate în ferestrele de proprietăți;
- Nu întotdeauna elemente de interfață care servesc aceleași sarcini sunt prezentate grupat;
- Efectul celor mai uzuale taste poate fi neașteptat. Spre deosebire de un editor de text, chiar și tastele corespunzătoare literelor din alfabet generează comenzi. De exemplu, tasta R repetă un pasaj scris, S inițiază un legato de expresie, N activează modul notare etc. Apoi, tastele cifrelor din tastatura alfanumerică activează alte comenzi decât cele din tastatura numerică. Din acest motiv interfața este foarte sensibilă la tastări eronate care au efecte nedorite, și, foarte probabil, având în vedere complexitatea aplicației, necunoscute utilizatorului. Faptul că de la primele cursuri utilizatorul află că poate anula efectul unei taste apăsată din neatenție cu Undo (Ctrl-Z) sau Escape îi conferă o siguranță și un sentiment de control al aplicației care îl încurajează în învățare. Totuși, este necesar ca studentul să fie conștient de acțiunile sale în calitate

awareness of their actions as a user and avoid random typing. The KLM sequences can thus be Ariadne's thread in the maze of menus.

In KLM's instructional approach, we are not interested in the duration of the R response – however negligible in current computational styles due to their high operating speed – but its content, which ensures the *resultiveness*<sup>2</sup> of each operation. Feedback cannot be omitted in the description of the steps, and students must get used to the fact that each command has an effect that is seen on the screen: in *Sibelius*, something may appear, the mouse pointer may change shape, or portions of the musical text may change colour. For example, *Sibelius* knows several types of selection, the visual response being different. In simple selection, objects appear blue (or green, orange, or purple, depending on the voice), whereas in system passage selection, they appear in a purple-coloured rectangle. In the case of graphic selection, the mouse pointer changes and takes the form of a cross, and the selected area is marked with a thin dashed line. In their demonstrations, the teacher will show these differences and explain their meaning. Students will get used to observing those signs to know if the application responded correctly to the command they launched.

In KLM's analysis of writing notes, there are R-type moments. The duration selection is marked by colouring the operated button (R1), and after typing the letter corresponding to the pitch, the note appears immediately on the scoreboard (R2). Therefore, the instructional sequence is more complex, containing and following this feedback, the formula is  $M_1M_2K_1R_1M_3K_2R_2$ . When the student checks the application's response, R-moments become milestones in the execution string. They or the teacher who helps them can thus identify and correct errors in time. For example, suppose the student who wants to enter a semiquaver accidentally types the 3 on the numeric keyboard. In that case, they will notice the colouring of the quaver and apply the correction by pressing the 2. As the student gains experience and skill, R-moments become less and less necessary, and they will tacitly omit them.

Type M operations are introduced at specific times, and the creators of the KLM method devised four precise rules for inserting them (Card, Moran, and Newell, 1983, 265). In the instructional algorithm, however, these moments will be more numerous, aiming to increase its *comprehensibility* (Landa, 1974, 11) by clarifying and making the student aware of the commands they give to the application. For this purpose, they can be accompanied, as we indicated, by R-type moments. Two of these rules are relevant to learning.

Rule two provides for eliminating M moments before each type if the series is part of a cognitive unit. Indeed, the editor will not prepare for writing every letter of the word *piano*. The formula is  $M K K K K K$  with one K typing for each letter of the word. Likewise, the musician seeing a C-E-G chord will intuit the two third intervals (M) and press C-3alphanumeric-3alphanumeric keys in an  $M K K K$  sequence. However, when designing and exposing the instructional algorithm for chords, the

de utilizator și să evite tastarea la întâmplare. Secvențele KLM pot fi astfel un fir al Ariadnei în labirintul meniurilor.

În abordarea instrucțională a KLM nu ne interesează durata răspunsului R – oricum neglijabil la modurile de calcul actuale datorită vitezei lor mare de operare –, ci conținutul său, care asigură *rezultivitatea*<sup>2</sup> fiecărei operații. Feedbackul nu poate fi omis în descrierea pașilor, iar studenții trebuie să se obișnuiască cu faptul că fiecare comandă are un efect, care le apare pe ecran: în *Sibelius* poate să apară ceva, indicatorul mouse-ului să își schimbe forma sau porțiuni din textul muzical să își schimbe culoarea. De exemplu, *Sibelius* cunoaște mai multe tipuri de selectare, efectul vizual fiind diferit. În cazul selecției simple, obiectele se colorează în albastru (sau verde, sau portocaliu, sau mov, în funcție de voce), pe când în cazul selecției de sistem, apar într-un dreptunghi colorat în mov. În cazul selecției grafice, se schimbă indicatorul mouse-ului, care ia forma unei cruci, iar zona selectată este marcată cu linie subțire întreruptă. În demonstrațiile sale, cadrul didactic va explica aceste diferențe și semnificația lor, iar studenții se vor obișnui să le urmărească, să observe dacă aplicația a răspuns corect comenzii pe care au lansat-o.

Și în analiza KLM a scrierii notelor există momente de tip R. Selecția duratei este marcată prin colorarea butonului acționat (R<sub>1</sub>), iar în urma tastării literei ce corespunde înălțimii, nota apare imediat pe portativ (R<sub>2</sub>). Prin urmare, secvența instrucțională este mai complexă, conținând și urmărirea acestui feedback, formula fiind  $M_1 M_2 K_1 R_1 M_3 K_2 R_2$ . Momentele R, în care studentul verifică răspunsul aplicației, devin borne în șirul execuției. El sau cadrul didactic care îl ajută pot astfel identifica și corecta la timp erori. Spre exemplu, dacă studentul care vrea să introducă o șaisprezecime tastează din greșeala tasta numerică 3, va observa colorarea optimii și se va corecta, apăsând tasta numerică 2. Pe măsură ce studentul dobândește experiență și îndemănare, momentele R devin tot mai puțin necesare, iar acesta le va omite tacit.

Operațiile de tip M se introduc în momente specifice, iar creatorii metodei KLM au conceput patru reguli precise privind inserarea lor (Card, Moran și Newell, 1983, 265). În algoritmul instrucțional însă, aceste momente vor fi mai numeroase, având scopul a spori *comprehensibilitatea* (Landa, 1974, 11) sa, explicând studentului comenzile pe care le dă aplicației. În acest scop ele pot fi însoțite, așa cum am arătat, de momente de tip R. Două dintre aceste reguli sunt relevante în învățare.

Astfel, a doua regulă prevede eliminarea momentelor M dinaintea fiecărei tastări în cazul în care seria face parte dintr-o unitate cognitivă. Într-adevăr editorul nu se va pregăti la fel de mult pentru scrierea fiecărei litere a cuvântului *piano*, ci doar o singură dată, înainte de a-l scrie. Formula este  $M K K K K K$ , cu câte o tastare K pentru fiecare literă a cuvântului. La fel, muzicianul care vede un acord Do-Mi-Sol va intui cele două terțe (M) și va apăsa taste C-3alphanumeric-3alphanumeric într-o secvența  $M K K K$ . Însă, atunci când va concepe și expune algoritmul instrucțional pentru acorduri, cadrul

<sup>2</sup> „Resultivity” is a quality descriptor of an algorithm that indicates its ability to lead to correct results in any situation. (Lewis, Horabin, and Gane, 1967).

<sup>2</sup> „Rezultivitatea” (*resultivity*) este un descriptor de calitate al unui algoritmul, care semnaleză capacitatea acestuia de a conduce în orice situații la rezultate corecte (Lewis, Horabin și Gane, 1967).

teacher will use the  $M M_1 K M_2 K M_{2bis} K$  formula, with moments  $M_1$ ,  $M_2$  and  $M_{2bis}$  serving to raise awareness of the introduction of the C note and the two ascending thirds.

Then, rule number one is to eliminate M-moments in front of fully anticipated operators. For example, the editor types Ctrl-C automatically without preparing for each of the two keystrokes (M K K). If a student does not know how to use the control key, the sequence operation will be explained in detail, becoming M K M K at a higher sequencing resolution. We see here the opportunity for differentiated education offered by approach offers. The instructor starts with an algorithm with minimal structure. If the students' feedback is negative, they will enrich it by inserting M operators, the physical sequence (Ks, Ps, Hs) remaining unchanged.

Beginner students need to acquire automatism as quickly as possible in such situations, eliminating additional M-moments. If Ctrl-C automatism is acquired almost instantaneously, practice is required to memorize and integrate the sequence in other situations, such as multiple selections. This selection assumes that after the first item has been selected, the other items are selected with clicks holding the *Control* key (e. g. for three items – M K P P P sequence). Even though this explanation is accompanied by a demonstration and immediately imitated by students, some forget and release *the Control* key after the second click. In this case, the teacher will add M-type operators, making the student aware of keeping the Control key pressed after each click (M K P M P M P). However, it is necessary to immediately foresee a lesson sequence in which the student will practice it until they become fluid in the habit of multiple selections.

Students who do not know English will find the application challenging. Not only do they have to memorize menus, but they also find it impossible to intuit and remember shortcuts or abbreviations that flow intuitively for English-speaking users. For example, the key that initiates the expression *legato* is S, short for its English name – *slur*. The solution is for the instructor to translate the terms they encounter in their explanations. Knowing that *slur* means *legato*, the student will more easily remember the letter introducing this sign. Thus, gradually and through repetition, students acquire the basic music terminology in English, gaining fluidity and autonomy in using menus. The visual command icons introduced with version 7 of the application are beneficial.

As Landa (1974, 33-36) and Kopstein (1974) observe, unlike computer systems that require the complete algorithm, human intelligence has the ability to imply information that is missing in the description of the algorithm. For this reason, the sequence of steps is simplified into an incompletely formalized algorithm, called by Landa *quasi-algorithm*, which holds the full power of a formal algorithm but in a simplified formula that can be learned more easily (Landa, 1974, 14).

Although the ellipse of steps characterizes the quasi-algorithm, the teacher must avoid ambiguity. Landa emphasizes that the specificity of the description is relative and depends on the student's level of knowledge, experience and imagination. We showed how, in the case

didactic va folosi formula  $M M_1 K M_2 K M_{2bis} K$ , momentele  $M_1$ ,  $M_2$  și  $M_{2bis}$  servind la conștientizarea introducerii notei Do și a celor două acorduri de terță în urcare.

Apoi, prima regulă prevede eliminarea momentelor M din fața operatorilor care sunt deplin anticipați. De exemplu, editorul tastează Ctrl-C în mod automat fără să se pregătească pentru fiecare dintre cele două tastări (M K K). Dacă un student nu știe cum se utilizează tasta control, i se va explica în amănunt operația, secvența devenind M K M K, printr-o explicare mai amănunțită a secvențierii. Observăm deschiderea către învățământ diferențiat pe care o oferă această abordare. Instructorul pornește cu un algoritm de structură minimală, iar dacă feedbackul studenților este negativ, îl va îmbogăți, inserând operatori M, succesiunea fizică (K-urile, P-urile, H-urile) rămânând neschimbată.

Important pentru studenții începători este ca în asemenea situații să dobândească automatismul cât mai repede, eliminând momentele de tip M suplimentare. Dacă în cazul tastării Ctrl-C automatismul este dobândit aproape instantaneu, în alte situații, precum cea a selecției multiple, este nevoie de exercițiu pentru memorarea și integrarea secvenței: după ce a fost selectat primul item, ținând tasta *Control* apăsată, se selectează cu clic și ceilalți itemi (pentru trei itemi – M K P P P). Chiar dacă această explicație este însoțită de o demonstrație și imediat imitată de studenți, unii uită și eliberează tasta *Control* după al doilea clic. În acest caz, cadrul didactic va adăuga operatori de tip M, conștientizând studentul să *mențină* tasta Control apăsată după fiecare clic (M K P M P M P). Este însă necesar să fie prevăzută *imediat* o secvență a lecției în care studentul să exerseze această secvență până când i se formează deprinderea selecției multiple.

Pentru studenții care nu cunosc limba engleză, aplicația este mai dificil de înțeles. Nu numai că trebuie să memoreze meniurile, dar le este imposibil să intuiască și greu să rețină anumite scurtături sau abreviații, care decurg intuitiv pentru vorbitorul de limbă engleză. De exemplu, tasta care inițiază un *legato* de expresie este S, prescurtare a denumirii sale în engleză, *slur*. Soluția este ca instructorul să traducă termenii pe care îi întâlnește în explicațiile sale. Știind că *slur* înseamnă *legato* studentul va reține mai ușor litera care introduce acest semn. Astfel, treptat și prin repetiție, studenții își însușesc terminologia de bază în limba engleză, câștigând fluiditate și autonomie în utilizarea meniurilor. De foarte mare ajutor sunt și iconițele comenzilor care au fost introduse odată cu versiunea 7 a aplicației.

Așa cum observă Landa (1974, 33-36), dar și Kopstein (1974), spre deosebire de sistemele informatice care au nevoie de algoritmul complet, inteligența umană are capacitatea de a subînțelege informații care lipsesc în descrierea algoritmului. Din acest motiv, succesiunea pașilor este simplificată într-un algoritm incomplet formalizat, numit de Landa *cvasi-algoritm*, care deține întreaga putere a unui algoritm formal, dar într-o formulă simplificată care poate fi învățată mai ușor (Landa, 1974, 14).

Deși cvasi-algoritmul este caracterizat prin omiterea unor pași, cadrul didactic trebuie să evite ambiguitatea. Landa subliniază că specificitatea descrierii este relativă și depinde de nivelul de cunoaștere a studentului, de experiența și imaginația sa. Am arătat cum în cazul

of keyboard input, the experienced user no longer needs moments  $M_2$  and  $M_3$ . Although they were taught the  $M M K M K$  sequence in the course, as they gain experience, it will be simplified to  $M K K$  and further to  $M_n (K K)$ , as the editor memorizes  $n$  signs before writing them. As a general rule, we have shown that in the design of the instructional algorithm, the teacher will insert several  $M$  moments of mental preparation that the student will “forget” as they acquire fluency in using the application. We are, therefore, dealing with a differentiated approach: The number of these  $M$  moments depends on the psychological assumptions of the student. In Sibelius’s case, we must consider the students’ knowledge of musical writing, primarily since the discipline is taught in the first year. Also, the general computer science training level is heterogeneous among students. Older students – the digital immigrants – may have serious gaps in their basic knowledge that need to be filled during the course or through remedial consultations.

## CONCLUSIONS

As they gain experience and fluidity, students will tend to replace instructional algorithms with quasi-algorithms customized according to acquired skills. Thus, algorithmization is a stage in mastering the use of an application at a higher level. Human intelligence can adapt the algorithm for new situations that have a degree of similarity to the initial one for which the learned algorithm was designed. This results in learning by example, in which the initially learned algorithm is interpreted elastically, being adapted to new situations. For example, students learn algorithmically the two methods of entering expression *legatos*: from the keyboard when writing notes or with the mouse. Then, learning that *legato* belongs to the class of lines, along with signs of *crescendo*, *decrescendo*, *trill*, *glissando*, and so on, they will intuitively adapt the two algorithms for all signs in this class.

In doing so, KLM proves its generalisation power by helping the user understand how, in the same formula of operator succession, only the nature of the order changes according to the intended purpose.

At a higher level, however, the user will interpret the app through metaphors that simplify its understanding but in which KLM is blurred. However, it can be reactivated if the user learns new functions either due to an update of the application or because the possibilities of its use become wider.

scrierii cu ajutorul tastaturii, utilizatorul experimentat nu mai are nevoie de momentele  $M_2$  și  $M_3$ . Deși i s-a predat la curs secvența  $M M K M K$ , pe măsură ce dobândește experiență, aceasta se va simplifica la  $M K K$  și, mai mult, la  $M_n (K K)$ , în măsura în care editorul memorează  $n$  semne înainte de a le scrie. Ca o regulă generală, am arătat că, în designul algoritmului instrucțional, cadrul didactic va insera mai multe momente de pregătire mentală  $M$  pe care studentul le va „uita” pe măsură ce dobândește fluentă în utilizarea aplicației. Prin urmare, avem de a face cu o abordare diferențiată, numărul acestor momente  $M$  depinzând de ipoteze psihologice ale studentului. În cazul programului *Sibelius* trebuie să avem în vedere cunoștințele de scriere muzicală ale studenților, mai ales că disciplina este predată în anul întâi. De asemenea, este foarte eterogen și nivelul de instruire în informatică generală. Studenții mai în vârstă – imigranții digitali – pot avea lacune serioase în cunoștințele de bază, care trebuie completate în timpul cursului sau prin consultații remediale.

## CONCLUZII

Pe măsură ce dobândesc experiență și fluiditate, studenții vor avea tendința de a înlocui algoritmi instrucționali cu cvasi-algoritmi personalizați, în funcție de deprinderile dobândite. Algoritmizarea este o etapă în stăpânirea la un nivel superior a utilizării unei aplicații. Inteligența umană are capacitatea să adapteze algoritmul noilor situații cu un grad de asemănare cu cea inițială, pentru care a fost conceput algoritmul învățat. Astfel, rezultă o învățare prin exemple, în care algoritmul inițial învățat este interpretat într-un mod elastic, adaptat noilor situații. De exemplu, studenții învață algoritmizat cele două metode de introducere a *legato*-urilor de expresie: din tastatură odată cu scrierea notelor sau cu ajutorul mouse-ului. Apoi, aflând că *legato* face parte din categoria liniilor (*lines*), împreună cu semnele de *crescendo*, *decrescendo*, *tril*, *glissando* ș. a. m. d., el va adapta intuitiv cei doi algoritmi pentru toate semnele din această categorie.

În acest demers KLM își dovedește puterea de generalizare, ajutând utilizatorul să înțeleagă felul în care, în aceeași formulă de succesiune a operatorilor, se schimbă doar natura comenzii, în funcție de scopul dorit.

La nivel superior, utilizatorul va interpreta aplicația prin metafore, care simplifică înțelegerea ei, dar în care KLM este estompat. Totuși, acesta poate fi reactivat dacă utilizatorul învață funcțiuni noi, fie datorită unei actualizări a aplicației, fie fiindcă posibilitățile de utilizare a acesteia se largesc.

## BIBLIOGRAFIE / REFERENCES

- [1] Al-Megren, S., Khalti, J., & Al-Khalifa, H. S. (2018). A Systematic Review of Modifications and Validation Methods for the Extension of the Keystroke-Level Model. *Advances in Human-Computer Interaction*, 2018, 1-26. DOI: 10.1155/2018/7528278.
- [2] Burns, M. T., Ritter, F. E., & Zhang, X. “L.”. (2016). Using Naturalistic Typing to Update Architecture Typing Constants. *Proceedings of ICCM 2016. 14<sup>th</sup> International Conference on Cognitive Modeling*, 169-174.

- [4] Card, S. K., Moran, T. P., & Newell, A. (1980b). The Keystroke-Level Model for User Performance Time with Interactive Systems. *Communications of the ACM*, Vol. 23, No. 7, 396-410. DOI:10.1145/358886.358895.
- [5] Card, S. K., Moran, T. P., & Newell, A. (1983). *The Psychology of Human-Computer Interaction*. Hillsdale, New Jersey, London: Lawrence Erlbaum Associates.
- [6] Clarke, T., & Spreadbury, D. (2009). *Sibelius 6 Handbook*. Avid Technology.
- [7] Embley, D. W., Lan, M. T., Leinbaugh, D. W., & Nagy, G. (1978). A Procedure for Predicting Program Editor Performance from the User's Point of View. *International Journal of Man-Machine Studies*, Vol. 10, No. 6, 639-650. DOI: 10.1016/S0020-7373(78)80025-X.
- [8] Holleis, P., Otto, F., Hussmann, H., & Schmidt, A. (2007). Keystroke-Level Model for Advanced Mobile Phone Interaction. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1505-1514. DOI: 10.1145/1240624.1240851.
- [9] Kopstein, F. F. (1974). What Is Algorithmization of Instruction? *Educational Technology*, Vol. 14, No. 10, 13-16.
- [10] Landa, L. N. (1974). *Algorithmization in Learning and Instruction*. New Jersey: Eaglewood Cliffs.
- [11] Lewis, B. N., Horabin, I. S., & Gane, C. (1967). *Flow Charts, Logical Trees and Algorithms for Rules and Regulations*. H. M. Stationery Office.
- [12] Metea, R. (2010). Sibelius 6 – Facilități în Compoziția muzicală/Sibelius 6 – Facilities in the musical composition. *ICTMF*, Vol. I, No. 2, 86-90.
- [13] Penciu, Traian. (2019). *Tehnoredactare muzicală. Suport de curs [Music Score Editing. Course support]*. Târgu Mureș: Ardealul.
- [14] *Sibelius 7.5 Reference Guide*. (2014). Avid Technology