VIRTUAL LEARNING ENVIRONMENTS: WHAT MAKES THEM EFFECTIVE

Conference Paper · November 2021

DOI: 10.21125/iceri.2021.0159

3 authors:

Thiago Pontes
Universidade Federal do Cariri

Guilhermina Miranda
University of Lisbon

Deborah Macêdo dos Santos
Universidade Federal do Cariri

VIRTUAL LEARNING ENVIRONMENTS: WHAT MAKES THEM EFFECTIVE

Pontes, Thiago B.[1,2], Miranda, Guilhermina L.[2], Santos, Deborah M.[1]

[1]*Universidade Federal do Cariri (BRAZIL)*

[2]*Instituto de Educação, Universidade de Lisboa (PORTUGAL)*

Abstract

The specialized literature in the field of early computer programming learning has analyzed the difficulties that students encounter, including abstract reasoning, problem-solving heuristics, and syntax errors. Studies indicate that the greatest difficulty for beginners is to combine the use of basic programming concepts and their effective use in coding. The use of virtual learning environments (VLE) in face-to-face teaching, eLearning and blending courses is a widely studied subject. However, there is little debate about the use of software engineering to design these virtual environments supported in psychological theories of learning and instruction. This article presents some results that were conceived from the practical application of agile software engineering methodologies, supported in the 4C/ID (Four Components Instructional Design) model that integrates the most conclusive results of two psychological theories about human cognition and learning: The Cognitive Load Theory and the Cognitive Theory of Multimedia Learning. The VLE was used to teach Python to computer-aided design software users. The 4C/ID model was used to try to reduce the difficulties in learning computer programming with Python, for students whose first option was not to learn computer programming, but who must learn it because it is part of the curriculum. The 4C/ID model adopts an approach to learning that is supported in the framework of information processing or cognitive psychology. In this approach, to test whether learning has occurred in a meaningful way, it is necessary to assess the knowledge acquired and also the ability to apply what was learned to new situations and problems. We used an experimental methodology with a quasi-experimental design with control and experimental groups within the same classes. We measured the acquisition and transfer of acquired knowledge and the mental effort. The mental effort scale was applied in two moments: The first time after applying the knowledge test and the second time after applying the transfer test. Mental effort is considered to be the total amount of processed cognitive control in which an individual is involved. An efficient instructional environment is one in which students are able to successfully solve the problems and learning tasks that are given to them with less perceived mental effort. With this procedure we tried to analyze which of the learning environments (conventional teaching method versus 4C/ID model) was more efficient, that is, where students obtained better results in the knowledge and transfer tests and perceived less mental effort. We used the t-Student for independent samples and ANOVA Kruskal-Wallis according to the assumptions of normality and homogeneity of variances. With the results obtained, we conclude that there is a difference in the perception of mental effort in favour of the experimental group; and that the experimental group obtained better results than the control group in tests of knowledge acquisition and transfer of learning. We concluded that the 4C/ID model is a good choice to develop efficient learning environments. That is why it is important to design online learning environments where students succeed with less mental effort. The 4C/ID model was precisely designed to improve the acquisition of the knowledge, skills, and attitudes involved in this complex learning, such as learning computer programming.

Keywords: Mental Effort, Software Engineering, Teaching Python, Virtual Learning Environments, 4C/ID Model.

1 INTRODUCTION

The challenges raised by teaching computer programming go beyond the barriers of time and technological evolution. Since its beginning, when the mathematician Dijkstra [1] introduced the subject in the book "A discipline of programming", the need to find teaching methods for beginners began to arise. At that time, it was a matter of logical-mathematical teaching to know how to operate with large computational machines, using mathematics as a language of communication.

The works by Soloway and Ehrlich [2] refer to the need to revise the curriculum for teaching programming aimed at beginners, with the "Learning to program =Learning to construct mechanisms and explanations". Boulay [3] in his work "Some difficulties of learning to program", managed to segment the difficulties of learning programming into specific areas. This author states that students often have difficulties in understanding the issues related to the execution of a program. Boulay [3] states that "it takes a long time to learn the relationship between a program on the page and the mechanism it describes." (1989, p. 290). He also stated that there must be a "notional machine", which simplifies the machine language so that all transformations in the program can be seen.

Winslow [4] started to use models to try to overcome the human difficulties in learning computer programming, in his work “Programming pedagogy: A psychological overview”. It addresses the programming challenges for beginners, grouped into three questions that reflect the difficulties presented by the students: (i) the relationship between understanding the real problem and the generation of the computational solution, (ii) the knowledge acquired in relation to praxis, (iii) and the relationships of the functional programming paradigm in comparison to to others paradigms.

Basically, there are two types of programming languages that are used in computer-aided design modeling software – CAD: visual programming (VPL) and textual programming (TPL). In this study, we approach the textual programming of computers, as it provides the user with greater breadth in relation to the possibilities for creating programs. The TPL chosen in this study is Python because, according to the studies by Villares and Moreira [5], there is a strong tendency to use Python as an embedded language in programming tools in about 40% of the studied software.

For Celani and Vaz [6], beginning students in programming languages can feel frustrated, given the complexity of understanding the syntax rules of each language. However, they claim that textual languages, such as Python, expand the possibilities of implementing new generative strategies, with a higher degree of complexity than visual languages which, generally, do not allow it.

To overcome the difficulties associated with the teaching methods, we tried to understand what alternatives we would have at our disposal. We resorted to the studies of Ludwig von Bertalanffy's General Systems Theory that, from the 1950s onwards, inspired so many others to create the basic concepts of instruction, learning and training, which resulted in the Instructional Systems Design (ISD) models, such as the ADDIE. According to van Merriënboer [7] this model divides the instructional design process into five phases: analysis, design, development, implementation, and evaluation.

The ADDIE model and other ISD models are comprehensive and work with formative assessments at all stages, and at the end of the process a summative assessment. Reigeluth [8] states that Instructional Design (ID) models are more detailed, that is, they maintain a more specific focus. For example, if we want to instruct in detail parts of a learning process that require complex problem solutions, we should use an ID model such as the Four Component Instructional Design (4C/ID) model. Complex learning is one that involves the integration of knowledge, skills and attitudes about a given domain of knowledge [9]. Below we will briefly describe the 4C/ID model.

1.1 Four Components to Instructional Design Model

The 4C/ID model aims to integrate knowledge, skills and attitudes and transfer them to real life [10]. This model also considers that real-life tasks are motivating to generate learning situations. Therefore, this model, when properly used, generates quality learning, that is, one that has positive effects on the acquisition and transfer of knowledge (Figure 1).

The model integrates results of experimental research carried out in the context of formal learning and in particular those deriving from the Cognitive Load Theory [11] and the Cognitive Theory of Multimedia Learning [12] [13].

To reduce the amount of information the students’ needs to remember to solve complex problems, the first of the four components, the Learning Tasks, are based on real-life problems. In these so-called authentic tasks, which integrate problems, projects and case studies, the student is invited to remember what they already know about the subject, that is, to test their previous knowledge, integrating skills and generating new knowledge [14]. Learning tasks, the core of the model, are classified into levels ranging from easy to difficult: tasks are grouped into task categories and within each category the student progresses from the easiest to the most difficult. At the beginning of solving each group or category of tasks, students receive a lot of support, and this is gradually withdrawn as they progress.

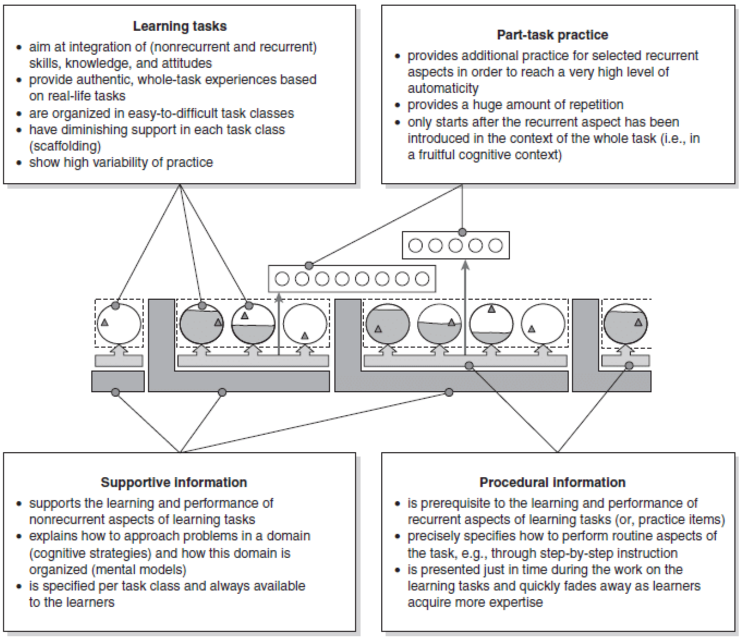

Figure 1. Basic Components of 4C/ID Model

The second component, Supportive Information, consists of information that helps the student to build a new point of view, based on previous knowledge and what they need to learn to work on the learning tasks. It is information that is always available, and that facilitates the articulation between theory and practice.

The Procedural Information, the third component of the model, is constituted by instructions on “how to do it”, which is information about what the student needs to learn in order to carry out the routine situations of learning tasks. This information should be accessed by students only when necessary, as a support to remember some details. It consists of algorithms, which bring with them a set of well-defined and ordered rules on how to perform a certain activity and can be accessed at any time.

Finally, the fourth component, Task Practice, can be understood as a test moment, which refers to the routine aspects of the tasks. However, instead of Supportive Information and Procedural Information, Practice on Tasks is presented as practical exercises, in which the student's autonomy is worked. Practice on tasks aims to consolidate learning by making certain components of tasks automated.

1.2 Virtual Learning Environment

An online instructional environment, created by the first author of this study, was developed in order to reliably meet what the 4C/ID model proposes. This tool was named "Elroy Learning Project" and was developed from the Bootstrap framework for the frontend layer, as it is a reuse solution that is easy to maintain and constantly updated. Currently, this framework is in version 4.x with HTML5 resource, Javascript, CSS3, Less and Sass, multiplatform. For the backend layer, the programming language in PHP 7.0 was used, and the database is MySQL.

The developed instructional environment was divided into two different access views: the teacher's and the student's.

The management of the functionalities of each section of the environment, as defined by Serra et al [15], is composed of the CRUD matrix as a reference that allows the description and analysis of the relationships between the activities of the processes and the information manipulated in the context of the business. In this matrix, each of its cells describes the actions that a process performs on an associated informational entity, which can be: Create, Read, Update and Delete.

1.2.1 AVA Instructor

The organization chart in Figure 2 details the conceptual map of the instructional environment in the instructor’s view. It shows that he/she can manage courses, invite students to courses, and manage their personal access information to the environment.

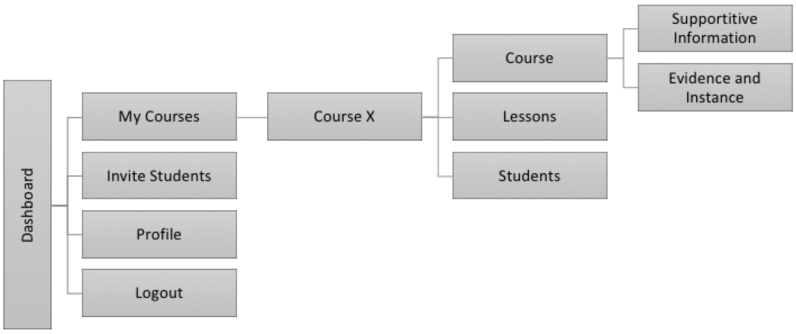

Figure 2. Organization chart of Elroy Learning Environment – Instructor's view

The Instructor can manage courses from the "My Courses" link so that he/she can edit course information such as period, key information, lessons, and students. In course management the instructor can manage information about the course, such as the course name, start and end date, and initial information what the course is about, such as curriculum and objectives.

Each course had three classes of lessons that followed the model proposed by Melo and Miranda [9] and that determine that:

1st. Lesson class – Students had to follow a set of three learning tasks (T1, T2 and T3) corresponding to a solved example (T1), a partially solved exercise (T2) and a complete task to be solved without help (T3).

2nd Lesson Class – Students had to complete a sequence of six learning tasks (T1-T6). The first three tasks corresponded to solved and partially solved examples and exercises; the last three tasks had to be solved without help.

3rd Lesson Class – Students solved a sequence of 16 tasks: the first learning task (T1) with partially solved example, the task (T2) with reduced support, and the others (T3-T16) without support with practical exercises.

1.2.2 Student at VLE

The organization chart in Figure 3 details the conceptual map of the instructional environment, now in the Student's view. It shows that the student can access the courses and manage their personal access information to the environment.

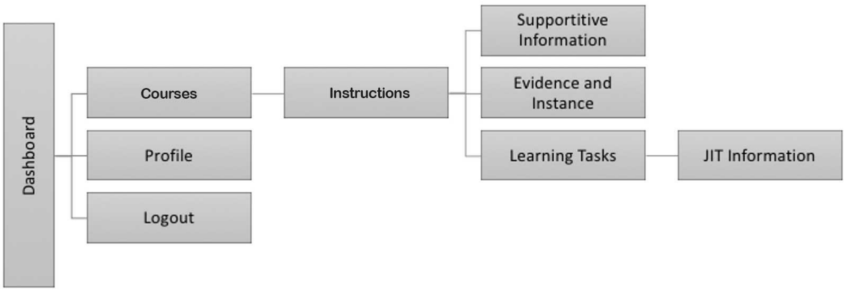

Figure 3. Organization chart of Elroy Learning Environment – Student's view

The first component of the 4C/ID model is the Learning Tasks. The first lesson has three task classes with one task (T) for each class (T1-T3) illustrated in Figure 4. Basic knowledge contents of Lists and Recursive Functions in Python were covered.

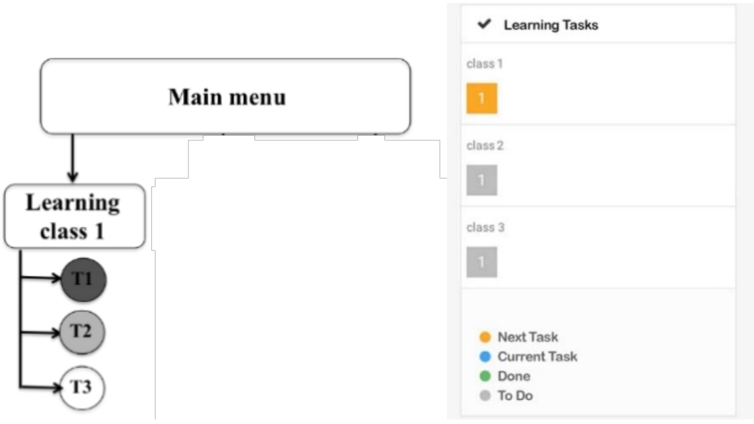

Figure 4. Task class structure vs Structure implemented in the VLE

The second component of the 4C/ID Model is the Supportive Information where the student must access the theory of this group of tasks in order to be led to use this information as an aid to making a connection between prior knowledge and what is needed to develop the tasks. In this experiment we used videos demonstrating how the Lists and Functions data structure for the Python programming language works.

Upon accessing the Supportive Information, the student is led to Demonstrations and Instances which, as the 4C/D model suggests, should be composed with the addition of examples in the context of the learning tasks. In this experiment, the students had access to the Demonstrations and Instances through the multimedia resource of explanatory videos, demonstrating the use of the data structure Lists and Recursive Function in Python.

The second lesson consisted of 3 task classes with one task for class 1 (T1), two for class 2 (T2-T3) and three tasks for class 3 (T4-T6). Contents of operations with Lists and Recursive Functions in Python were discussed.

The third lesson was also composed of 3 task classes with one task for class 1 tasks (T1), one task for class 2 (T2) and fourteen tasks for class 3 (T3-T16), illustrated in Figure 5. Contents of abstract representations with the use of Lists or graphical representations with Recursive Functions in Python were addressed.

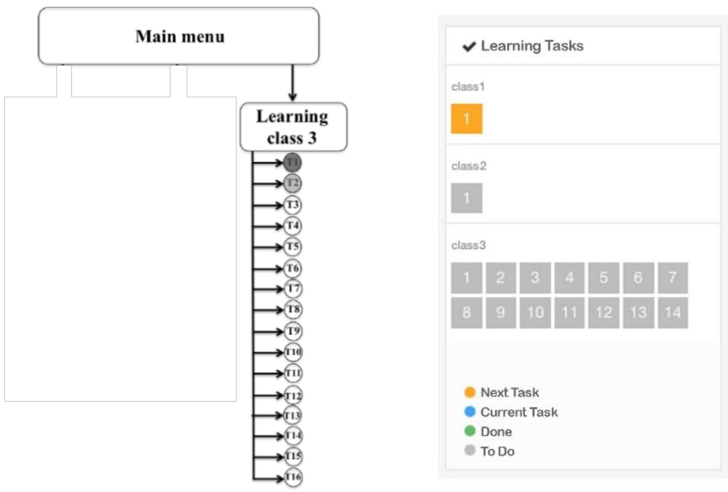

Figure 5. Structure of task classes III vs Structure implemented in VLE

In Figure 6 we present the screenshot of the abstract representations task using Lists and Recursive Functions.

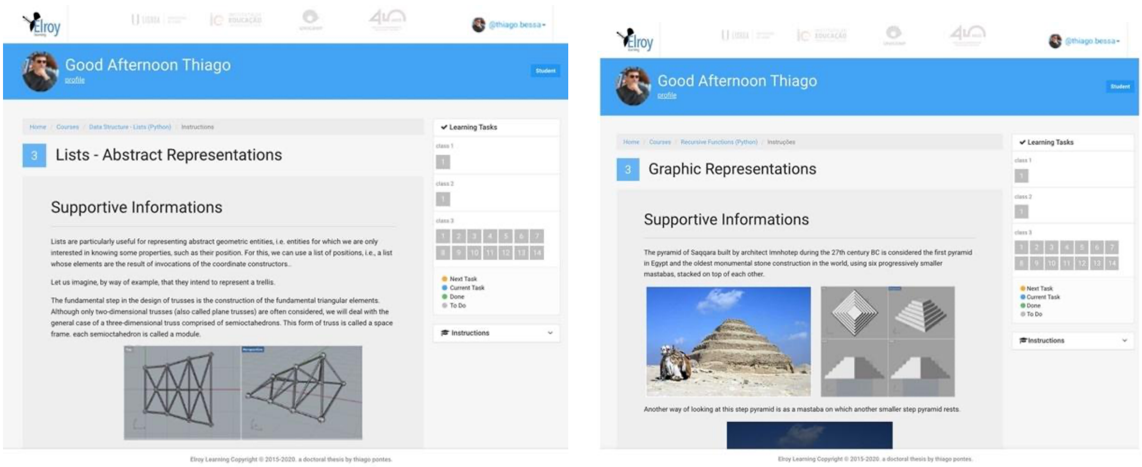

Figure 6. Screenshot of abstract representations task using Lists and Recursive Functions

At any time, the student accesses the instructional environment or if he/she have navigated to the home page, the environment indicates which learning task the student should recommence.

After completing all the learning tasks, the student is taken to the homepage of the environment, which gives him/her a message of completing the tasks and filling out the post-test surveys.

2 METHODOLOGY

We used a quantitative research methodology, with a quasi-experimental design, using control and experimental groups within the same class. We worked with Portuguese and Brazilian samples, from the Degree in Architecture, during the 2017/2018 academic year at the Instituto Superior Técnico (IST) in Lisbon, and a class of Architecture and Urbanism from the Faculdade de Engenharia Civil e Arquitectura, da Universidade de Estado de Campinas (FEC/UNICAMP).

The experimental factor or independent variable was the teaching method which assumed two values: the 4C/ID model and the conventional method; the dependent variables were the results of these teaching methods on: (a) the knowledge acquired by the students, (b) the transfer of this knowledge from computer programming to similar tasks, and (c) the perceived mental effort.

2.1 Mental Effort Scale

To measure the mental effort perceived by the students, we used the Subjective Cognitive Load Measurement Scale (SCLM), developed by Paas [16]. It was applied after the participants had performed two tests: one for Knowledge Acquisition and another for Transfer. SCLM is a one-dimensional scale that consists of a classification in a numerical representation from 1 (very very low mental effort) to 9 (very very high mental effort). The authors Alves et al. [17] suggest that measuring the amount of mental effort employed in performing a task allows for the improvement of the development of instructional tasks, promoting better learning.

2.2 Knowledge Acquisition and Transfer Assessment Tests

Preparing a test that assesses the level of knowledge in any domain is always a challenge for a teacher and even for a researcher. Assessing basic computer programming knowledge is no different. Costa and Miranda [18] concluded, in a meta-analytic study on programming languages, that "it is difficult to compare the results of investigations that aimed to improve the programming skills of students who were at an early stage of learning, not only because different programming languages were used, but also because the evaluation made used different measurement instruments." (p. 67).

The knowledge acquisition and transfer assessment tests used in this study were tailor-made to assess the knowledge acquired by students and their ability to transfer it to similar tasks. These tests were applied to students from both groups: those who were taught by the conventional method and those taught by the 4C/ID model. Each group was divided into two subgroups defined at random, in which group "A" received the formation of the syllabus "Recursive Functions" with the conventional method, and group "B" with the method based in the 4C/ID model. In a second moment, the content of

"Lists" was applied to group "B" with the conventional method and to group "A" with the 4C/ID model method, to avoid 'parasitic' effects of the experimental treatment, as can be seen in Figure 7.

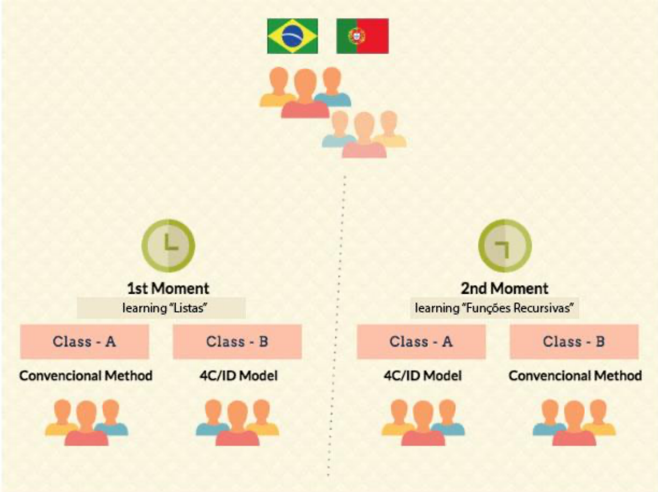

Figure 7. Infographic with the experimental procedures adopted

3 RESULTS

The main objective of this research work was to develop an online instructional environment capable of enabling the creation of learning tasks based on the 4C/ID model, and to evaluate the effects of this environment on the perceived mental effort and on the acquisition and transfer of programming knowledge.

The majority of students (90.9%) claimed, in a questionnaire applied before the start of the intervention, to have difficulty in creating codes, stating that this was due to a lack of understanding of the mathematics involved in solving problems and not mastering the rules of language under study. But about 83 percent of students (82.9%) said they were motivated to acquire knowledge in computer programming language Python.

We understand that to meet the needs of these students, with regard to the complex learning involved in this investigation, it would be necessary to design a learning environment that would support them to overcome their difficulties and that would translate into positive and significant results in the acquisition and transfer of knowledge in which students perceived less mental effort.

3.1 Perceived Mental Effort

Perceived mental effort is the total amount of cognitive processing controlled by an individual while performing a task. This construct was evaluated in two moments: at the end of the knowledge acquisition test and at the end of the transfer test.

For both tests, we verified that there were changes between the results of means when comparing the control group and the experimental group, always in favor of the experimental group.

In validating the assumptions of normality and homogeneity of variances, we obtained a normal distribution for Group 1 - Lists, as the variances were homogeneous ($p>0.05$), in both tests (acquisition and transfer) and in both groups (control and experimental). We applied Student's t-test for independent samples.

For Group 2 – Functions, we verified that the data presented a non-normal distribution ($p\leq0.05$), with a non-homogeneous distribution of variance (p-value=0.02 and p-value=0.47 for the knowledge acquisition and transfer tests respectively). We applied the non-parametric Kruskal-Wallis test.

Analyzing Table 1 for Group 1 – Lists, we observed that the means of the control group represent a relatively high mental effort, and the experimental group a medium mental effort. In this case, there was an improvement in the perception of mental effort in favor of the experimental group. The t test obtained a value of $t(62)=2.556$, which allows us to conclude that there is a statistically significant difference in the perception of mental effort for the List knowledge acquisition test, in favor of the experimental group. This means that students in the experimental group who learned Lists using the 4C/ID model experienced less mental effort.

For the knowledge transfer test, descriptive statistics shows that, on average, there was no change in the perception of mental effort. Levene's test was not statistically significant (p=0.814) and the t test presented a value of t(62)=1.339 with a p=0.185. These results are not statistically significant.

Table 1. Results of Mental Effort for Lists – Group 1.

		Control Group	Experimental Group	Levene's Test	t-test	
				Sig.	t	Sig.
Knowledge acquisition	Mean	6,41	5,35	,709	2,556	,013
	Std. Deviation	1,581	1,641			
	Minimum	2	2			
	Maximum	9	8			
Transfer of Knowledge	Mean	7,51	7,09	,814	1,339	,185
	Std. Deviation	1,207	1,240			
	Minimum	4	5			
	Maximum	9	9			

For Group 2 – Functions, in the knowledge acquisition test it was observed that the control group had a mean=5.30 (SD = 1.363), that is, a median mental effort, and the experimental group had a mean=4. 00 (SD = 1.897), which means, a small mental effort. In this case we noticed an improvement in the perception of mental effort between the control and experimental groups.

For the knowledge transfer test of Group 2 - Functions, we found that, on average, there was a change in the perception of the mental effort employed, in favor of the experimental group (grade 7 - high mental effort) compared to the control group (grade 8 - very high mental effort).

With the Kruskal-Wallis ANOVA for Group 2 – Functions, it was possible to conclude that both for the acquisition test and for the knowledge transfer test there was a change in the perception of mental effort in favor of the experimental group. We rejected H0 and accepted H1.

Table 2. *Results of the Mental Effort for Functions – Group 2.*

		Control Group	Experimental Group	ANOVA de Kruskal-Wallis
				Sig.
Knowledge Acquisition	Mean	5,30	4,00	,007
	Std. Deviation	1,363	1,897	
	Minimum	2	1	
	Maximum	7	8	
Transfer of Knowledge	Mean	7,87	6,63	,001
	Std. Deviation	,968	1,462	
	Minimum	6	3	
	Maximum	9	9	

We concluded that the 4C/ID model was advantageous for students in terms of the perception of the mental effort spent in carrying out the Lists and Functions learning tasks. Students who used the 4C/ID model perceived less mental effort in solving the proposed tasks than their peers who were taught with the conventional method

3.2 Knowledge Acquisition and Transfer of Knowledge

The hypothesis formulated was that students who used the 4C/ID instructional model to learn to program (Group 1 - Lists and Group 2 - Functions) would have better results than students taught by the conventional method in the Acquisition and Transfer tests of Knowledge.

We found that in both situations the classifications obtained by students in the experimental group were higher than those obtained by students in the control group.

The best result was obtained by Group 2 – Functions, in the knowledge acquisition test, with an average of 18,317 for the experimental group compared to 12,478 obtained by the control group, on a scale of 0 to 20 values. The results of this Group 2 – Functions, presented a non-normal distribution ($p \leq 0.05$), although the homogeneity of the variance ($p > 0.05$) was verified, which made us choose the t-Student test for independent samples. Both for the knowledge acquisition test and for the transfer test, the results of the mean values were statistically significant. With these results, we concluded that there was a better average performance of students in favor of the experimental group, and we accepted our H1, rejecting the H0 hypothesis.

Group 1 – Lists presented a non-normal distribution of data, however there was homogeneity of variance. Therefore, we decided to apply the t-Student test for independent samples in which the differences observed between the mean values in the knowledge acquisition test were statistically significant: $t(62) = -5.767$, and $p \leq 0.05$. These results were not observed for the knowledge transfer test, since the results obtained with the t-Student test for independent samples were not statistically significant: $t(62) = -1{,}500$ and a $p=0.139$. With these results, for the knowledge transfer test, we accept the null hypothesis. However, we also concluded that for the knowledge acquisition test there was a more positive average performance of students in favor of the experimental group, and in this case, we accepted H1.

In summary: We conclude that the results of inferential statistics allow us to accept H1 for the knowledge acquisition tests in favor of students who used the 4C/ID model, either for learning Lists or for learning Functions.

Regarding knowledge transfer for similar tasks, in favor of students who used the 4C/ID model we accepted H1 for the knowledge transfer of the Functions content, and we had to accept H0 for the List knowledge transfer test, which means that both students taught using the conventional method and students taught using the 4C/ID instructional model obtained similar results in this test.

To reinforce these inferential statistics results, we present in Table 3 descriptive statistics: results obtained by students from both groups: experimental and control. It is verified that the results are always favorable for the experimental group in the three variables measures: knowledge acquisition, knowledge transfer and perception of mental effort.

Table 3. Caption for the table.

			Control Group	Experimental Group
Knowledge Acquisition	Group1 - Lists	Perceived Mental Effort	6	5
		Test Score	13,6	18,3
	Group2 - Functions	Perceived Mental Effort	5	4
		Test Score	12,5	18,3
Transfer of Knowledge	Group1 - Lists	Perceived Mental Effort	7	7
		Test Score	9,5	11,9
	Group2 - Functions	Perceived Mental Effort	8	7
		Test Score	9,2	12,5

4 CONCLUSIONS

We think that the main contribution of this research lies in supporting the hypothesis that the use of instructional methodologies that help students to build knowledge of complex learning is verified, and that the 4C/ID model is a suitable model for this type of learning.

More specifically, we found that an instructional model centered on learning tasks, based on cognitive theories that guide the use and creation of multimedia learning objects, which meet the cognitive limitations of the learners, facilitates learning the Python programming language for the elaboration of drawings aided by algorithm, with the use of CAD software.

It was also possible to verify how the 4C/ID model allows to accurately instruct problems that require complex solutions. We believe that this is due to its analytical and precise bent, which translates into positive results in the ability to acquire and transfer knowledge, with a lower perception of mental effort on the part of the students. These are the three characteristics that determine the efficiency of an instructional environment.

REFERENCES

[1] Dijkstra, E. W. (1976). A discipline of programming. Eindhoven: Prentice-Hall.

[2] Soloway, E., & Ehrlich, K. (1984). Empirical studies of programming knowledge. IEEE Transactions of Software Enginnering, 10(5), 595-609.

[3] Boulay, B. D. (1989). Some difficulties of learning to program. Journal of Educational Computing Research, 2(1), 283-299.

[4] Winslow, L. E. (1996). Programming pedagogy - a psychological overview. ACM SIGCSE Bulletin, 28(3), 17–22.

[5] Villares, A. B. do A., & Moreira, D. de C. (2017). Python on the landscape of programming tools for design and architectural education. XXI Congreso de la Sociedad Ibero-americana de Gráfica Digital, 207–211.

[6] Celani, G. V., & Vaz, C. E. V. (2012). CAD scripting and visual programming languages for implementing computational design concepts: a comparison from a pedagogical point of view. International Journal of Architectural Computing, 10(1), 121–137.

[7] van Merriënboer, J.J.G. (1997). Training complex cognitive skills: a four-component instructional design model for technical training. Englewood cliffs. New Jersey: Educational Technology Publications.

[8] Reigeluth, C.M., Stein, F.S. (1983). The elaboration theory of instruction. Instructional design theories and models: an overview of their current states. Hillsdale: Lawrence Erlbaum.

[9] Melo, M., & Miranda, G. L. (2015). Learning electrical circuits: The effects of the 4C/ID instructional approach in the acquisition and transfer of knowledge. Journal of Information Technology Education: Research, 14, 313- 337.

[10] van Merriënboer, J. J., Kirschner, P. A., & Kester, L. (2010). Taking the load off a learner's mind: instructional design for complex learning. Educational Psychologist.

[11] Sweller, J., Ayres, P., & Kalyuga, S. (2011). Cognitive load theory. New York: Springer.

[12] Mayer, R. (2001). Multimedia learning. New York: Cambridge University Press.

[13] Mayer, R. E. (2005). The Cambridge handbook of multimedia learning. New York: Cambridge University Press.

[14] van Merriënboer, J. J., & Kester, L. (2005). The four-component instructional design model: multimedia principles in environments for complex learning. New York: University Press.

[15] Serra, N., Gonçalves, A., Serra, J., & Sousa, P. (2008). Planeamento estratégico no contexto de arquitectura empresarial. Revista Ibérica de Sistemas e Tecnologias de Informação, (2), 1–14.

[16] Paas, F., & Van Merriënboer, J. J. G. (1993). The efficiency of instructional conditions: An approach to combine mental effort and performance measures. Human Factors, 35, 737-743.

[17] Alves, M. V. C., Modesto, J. G., Lima-Rossetti, D., Lanini, J., & Bueno, O. F. A. (2017). As dimensões da carga cognitiva e o esforço mental in Revista Brasileira de Psicologia, 04(01), 2–16. Disponível em: http://revpsi.org/wp-content/uploads/2018/03/Alves-et-al.-2017-As-dimensões-da-carga-cognitiva-e-o-esforço-mental.pdf

[18] Costa, J. M., & Miranda, G. L. (2017). Desenvolvimento e validação de uma prova de avaliação das competências iniciais de programação. RISTI - Revista Iberica de Sistemas e Tecnologias de Informação, 2017(25), 66–81. doi:10.17013/risti.25.66-81