



Mind the gap: Impact on learnability of user interface design of authoring tools for teachers



David Roldán-Álvarez^a, Estefanía Martín^{b,*}, Manuel García-Herranz^a, Pablo A. Haya^c

^a Universidad Autónoma de Madrid, Ciudad Universitaria de Cantoblanco, 28049 Madrid, Spain

^b Universidad Rey Juan Carlos, c/Tulipán s/n, 28933 Móstoles, Madrid, Spain

^c Instituto de Ingeniería del Conocimiento, Universidad Autónoma de Madrid, Ciudad Universitaria de Cantoblanco, 28049 Madrid, Spain

ARTICLE INFO

Article history:

Received 7 January 2015

Received in revised form

27 April 2016

Accepted 27 April 2016

Available online 26 May 2016

Keywords:

Authoring tools and methods

Human-computer interface

Evaluation of CAL systems

ABSTRACT

Few teachers include information and communication technology in the classroom, despite their potential for increasing attention and motivation for students. Educational authoring tools are intended to turn teachers into designers and deliverers of technology-enhanced educational content, and increasing the adoption of these tools is a key element for speeding up this transformation. This paper emphasizes the importance of learnability for preventing rejection or abandonment by of such an authoring tool, and how acceptance is deeply affected by the interaction paradigm and the creation metaphor used in the tool. We present an analysis comparing two design paradigms: the widespread menu-based and choice-guided interaction paradigm versus a consistent metaphor with direct manipulation. The latter was implemented in DEDOS-Editor, a novel authoring tool that allows the creation of diverse educational activities that can be performed on different devices, such as PCs, digital blackboards, tablets, and multitouch surfaces. An experimental study shows the tremendous impact that interface choices have on the tool's learning curve. The results provide the first mapping of the choice of a direct-manipulation interface and its effect on the learning curve's entry point, as well as a consistent interaction metaphor with smoother and fast-growing learning curves. This allows users to complete more tasks and gain more knowledge through experience, in contrast to menu-based interfaces. The initial use of the tool is thus made easier for users with no experience or information about the tool, and the advantages of experience and expertise in facing new challenges are facilitating. This work also highlights the appropriateness of learning curves as a tool for measuring learnability.

© 2016 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

Information and Communication Technology (ICT) offers many potential possibilities to teachers for creating educational activities that students can perform in the classroom. "The Survey of Schools: ICT in Education" (European Union, 2013) sheds light on the use of ICT in the classroom. First, most students think that using computers for learning is interesting, they are motivated to learn with them, and they are used to performing tasks with electronic devices at home. Second, school heads and teachers agree about the relevance of ICT use in different learning activities and acknowledge that technologies motivate students. However, despite the significant differences between EU countries regarding ICT and electronic equipment use in the classrooms, "on average

across the EU countries [...], only between 20 and 25% of students are taught by digitally confident and supportive teachers having high access to ICT and facing low obstacles to their use at school." This information points out the need for teachers who are confident and supportive in effectively using ICT infrastructure and exploiting their potential. In agreement with Levy and Ben-Ari (2007), we believe that this poor use of technology in the classroom is mainly due to teacher limited knowledge of existing technological resources. This problem is worsened by the severe scarcity of educational tools (although this is slowly being solved by the inclusion of tablets in classrooms), as well as the rigidity of available tools, which do not comply with teacher needs in most cases (Hutchful et al., 2010).

One approach to this issue is to steer development of applications specifically designed to support teaching in a particular area of knowledge. However, developing a final application is a daunting task involving a costly and time-consuming process of analysis, planning, coding, and testing (and later support) to target a particular need. Targeting needs poses a new problem of applications not being adaptable enough to teacher needs (Pelgrum,

* Corresponding author.

E-mail addresses: david.roldan@uam.es (D. Roldán-Álvarez), estefania.martin@urjc.es (E. Martín), manuelgarciaherranz@uam.es (M. García-Herranz), pablo.haya@iic.uam.es (P.A. Haya).

2001). In this situation, technical expertise is needed to add new requirements to a certain application, which will require teachers to have a strong technological background or technical support. Usually, teachers do not have technical skills, and since having technical support is commonly expensive, teachers will end up abandoning ICT (Buabeng-Andoh, 2012).

Putting the creative power in the teacher hands requires authoring tools with which teachers can continuously and smoothly improve the creation process. In this study, we aim to highlight the factors affecting the learnability of an authoring tool, which is understood as “the capability of the software product to enable the user to learn its application” (ISO/IEC 9126-1, 2000). As Carroll et al. (2003) suggested, we strongly believe that learnability is one of the critical factors for making teachers adopt a specific technology. In designing an authoring tool, we should improve the familiarity (“the extent to which a user’s knowledge and experience in other domains can be applied when interacting with a new system”), the predictability (“support for the user to determine the effect of future action based on past interaction history”), and the generalizability (“support for the user to extend knowledge of specific interaction within and across applications to other similar situations”) (Dix et al., 2004). In this sense, learnability can be optimized by measuring the teacher’s performance in designing educational activities over time, which can allow us to identify flaws or characteristics of the developed tool and improve it to ease the learning curve. Teachers show different learning curves when using different authoring tools that are affected by both the user interface design (Nicholas, 2009) and the teacher’s background skills and prior knowledge (Ainsworth et al., 2004). Thus, design decisions will affect how easy teachers find the tool upon first use, how fast they acquire skills, and how competent they become after using the tool for a fair amount of time. These effects can in turn be observed in the learning curve of each user.

We were thus motivated to design and implement DEDOS-Editor, a novel authoring tool meant to provide a consistent design metaphor to smooth the learning curve for teachers. Our approach capitalizes on the creativity and expertise of teachers in the creation process, as well as the group dimension of the educational system by allowing teachers to access, use, and build over content developed by other teachers. In the design of DEDOS-Editor, we focused on improving learnability through a direct-manipulation interaction style (Shneiderman, 1983) and a consistent interaction metaphor (Nielsen, 1989) for the creation of educational activities. We started with the following questions:

- What are the most popular activities designed by teachers with authoring tools?
- Is learnability affected by the interaction style? What design paradigm improves learnability?
- Which factors impact the learning curve of a tool?

To test our hypotheses, we performed a controlled experimental study with bachelor’s students aged 21–30 years old in primary education and early childhood education (future teachers) at Universidad Rey Juan Carlos, Spain. The experiment compares DEDOS-Editor with an authoring tool that can be seen as a fair example of those available. After reviewing the state of the art, we chose JClick-Author (XTec, 2016), which is the most popular educational creation tool for personal computers and digital blackboards in Spain. Both tools were compared to collect information about their learnability, which helps us to comprehend teacher struggles when first facing a new tool in their classrooms and the main reasons they stop using them.

This paper is structured as follows. In Section 2, we analyze the current educational authoring tools and whether they support collaborative learning using different devices. Section 3 discusses

how design principles affect the learning curve and which design principles we have followed. Sections 4 and 5 detail the experimental study, the characteristics of the participants, the methodology, and the results. Finally, Sections 6 and 7 summarize the discussion and conclude the work.

2. Related work

The most frequent ICT-based activities in the European Union are related to the preparation of educational activities, creating digital resources, or using a virtual learning environment (European Union, 2013). However, the resources most used are multimedia tools such as PowerPoint, exercise software, online quizzes/tests, and broadcasts (podcasts, Youtube), among others. Therefore, available tools and devices are not being fully exploited. We strongly believe that this is due to developers overlooking the importance of the learning curve of certain applications. Learning applications should consider short-term efforts and long-term limitations (Papert, 1980) for users to gain knowledge of the application quickly while avoiding boredom for experts. We go deeper into this topic in Section 3.1.

In exercise software, teachers use special authoring tools to create interactive learning activities for students to use through PCs or digital blackboards. In addition, teachers can easily share the created contents with these applications, receive feedback from other teachers, and improve the learning content provided to the academic community (Su et al., 2005). We looked at the design paradigms of the most popular examples that had any kind of evaluation behind them (Venugopal et al., 2005), and we have found that the menu-based interaction style is widespread (see Table 1).

A prototypical example may be JClick (XTec, 2016), which allows for different types of educational activities, such as puzzles, associations, text activities, and crosswords. Activities are grouped in projects. Its interface is based on a menu, and it has four main tabs: “Project” for defining general features; “Media Library,” where the teacher must provide all the multimedia contents such as images, audio, or video; “Activities,” where the teacher selects the next activity to be created or updates previously defined ones; and “Sequences,” where the order to accomplish the activities is specified. A different set of options is displayed, depending on the type of activity selected. Hot Potatoes is another project with a similar approach (Half-Baked Software Inc. and University of Victoria Humanities Computing and Media Center, 2013). It requires previous knowledge related to web design or the use of image manipulation programs if teachers want to include images of different sizes (e.g., GIMP, Photoshop, or Paint).

Smart Notebook (SMART, 2013) uses an alternative approach where activities are designed by dragging and dropping visual elements or by directly painting on the screen. However, the activity creation is not consistent across different types of activities, since each type must be selected in a menu and different options appear in each element to configure the corresponding activity. In the same line of dragging and dropping, the Mouse Mischief add-on (Moraveji et al., 2008) harnesses the potential of the PowerPoint interaction paradigm to turn it into a simple but powerful authoring tool. However, it has a small variety of activities.

Authoring exercise functionality can also be found embedded in tools allowing teachers to create course-based learning environments, where activities are designed to be executed as part of a course. A fair representative of this class of tools is Microsoft LCDS (Microsoft Learning, 2013), which helps teachers in creating learning environments that will be executed as computer-based training applications by specifying their own pedagogical principles. Other projects that focus on authoring courses are Xerte

Table 1
Comparison of educational authoring tools regarding their interaction style and type of activities.

Authoring tool	Interaction style	Educational activities type
ARIES	Hierarchical object paradigm/Menu	Augmented reality simulations
Bookbuilder	Frame paradigm/Menu	Creating courses
Cognitor	Tagging paradigm/Menu	Creating courses
DEDOS-Editor	Card-based paradigm/Drag and drop	Simple and multiple choice, pair matching, and mathematical activities
Hot Potatoes	Hypermedia linkage paradigm/Menu	Simple and multiple choice, pair matching, crosswords, order words, fill in the gaps
JClic	Hypermedia linkage paradigm/Menu	Simple and multiple choice, pair matching, exploration, puzzles, fill in the gaps, memory games, crosswords, alphabet soup, identifying sounds
JeLSim Toolkit	Frame paradigm/Menu	Physics and Math Simulations Simple and multiple choice
LAMS	Icon-based paradigm/Drag and drop	Simple and multiple choice, text answers
Microsoft LCDS	Icon-based paradigm/Menu	Simple and multiple choice, essay
Mouse Mischief	Icon-based paradigm/Menu and drag and drop	Simple and multiple choice, fill in the gaps
Quandary	Hypermedia-linkage paradigm/Menu	Creating courses, simple and multiple choice
REDEEM	Frame paradigm/Menu	Simple and multiple choice, creating courses
Smart Notebook	Card-based paradigm/Drag and drop	Simple and multiple choice, pair matching, text answer, identifying sounds, order words
Xerte	Scripting paradigm/Menu	Creating courses, simple and multiple choice, pair matching, fill in the gaps

(University of Nottingham, 2008), REDEEM (Ainsworth et al., 2004), Bookbuilder (CAST, 2006), Cognitor (Neto et al., 2006), and Quandary (Half Baked Software, 2003).

Such applications are based on configuring a course across multiple menus and forms. In these tools, activities are organized according to a course structure rather than a project structure. Therefore, the output combines educational material with activities in which content creation occupies a significant portion of the design process. Interactive activities are designed in a similar way to JClic or Hot Potatoes. However, being more ambitious in scope, they tend to focus on professional content creators rather than on teachers with time constraints and a basic computer background in most cases. This impedes the learnability of the tool, making help mechanisms and virtual assistants critical for the tool's adoption. For instance, REDEEM assists in the creation of a learning environment by guiding the user through a question and answer process. This allows teachers to design complex tasks with fairly simple actions, although it may be a time-consuming process. Likewise, LAMS also facilitates the definition of course structures by using direct manipulation, but being oriented to multimedia content creation, activity types are limited to questions and answers (LAMS International, 2004).

One relevant aspect usually forgotten when analyzing learnability is variation in teacher computer skills. An interesting approximation that accounts for these differences is JeLSIM toolkit (Thomas and Milligan, 2004), a software suite for easily creating and customizing physics and math simulations. The toolkit shows different interfaces according to the user's computer proficiency. To create identical simulations, a Java programmer has access to a programming interface, whereas a teacher without proper programming skills can use a graph-based graphical interface to simplify the creation and simulation process. The variety of user interfaces is very interesting from the point of view of learnability, although it makes it more complicated to jump from one level of expertise to the next, and the implementation might have benefited from including an intermediate proficiency level.

Apart from the interaction mechanism, another issue is that few tools offer the possibility of creating activities that multiple students can solve at the same time and on the same device. For example, SMART Notebook activities are intended to be performed by a single student, despite the fact that more than one can interact with the digital blackboard to do the activity. If teachers want students to perform the same activity in different workspaces, they should replicate the activity for each student. The groupware approach of Mouse Mischief allows interaction with multiple pointers in a single display, although its approach is

constrained to the use of mouse devices. LAMS provides the possibility of performing collaborative activities, but it is online collaboration where each user performs activities on a separate device.

In summary, most tools seem to overlook the importance of their learnability. All of them partially ignore the necessity of providing a coherent and consistent mechanism for creating different types of activities, and in our experience, many of them provide clumsy user interfaces that do not reflect that somebody has to design an activity before a student performs it. Furthermore, the very few tools have been evaluated, like REDEEM or ARIES (Wojciechowski and Cellary, 2013), have focused on the student's performance in accomplishing the activities rather than on the teacher's performance in designing them. In the following section, we introduce the importance of learnability and the design principles we have followed to improve learnability in DE-DOS-Editor.

3. Learning curve and design principles

The design of computer-supported pedagogically sound group-learning activities still remains a complex task that requires both technological and pedagogical skills and knowledge. To manage these multidisciplinary needs, it is common for pedagogical teams to join efforts with technological ones, which leads to a completely new set of problems from a designer's point of view. These problems can be summarized in two main design requirements: flexibility and ease of use.

3.1. Learning curve

It is necessary to consider the learning curve when a computer tool is designed. As Repenning and Ioannidou (2006) pointed out, based on the optimal flow of motivation in learning (Csikszentmihalyi, 1991), anxiety occurs when challenges exceed user abilities, while boredom arises if skills are above the challenges that can be tackled with the tool. That is, limited knowledge must suffice to do simple things, while advanced knowledge should gradually lead to increasing construction power. This means paying particular attention to short-term efforts and long-term limitations (Papert, 1980) to reduce newcomer anxiety while avoiding expert boredom. Therefore, it is important to minimize both the effort required to begin using a tool as well as its long-term limitations, which would allow users to tackle increasingly complex challenges as they gain knowledge and experience.

Mueller et al. (2008) identified certain factors that kept teachers from fully embracing computers as teaching tools: a varied set of comfort levels with computers, strong doubts about computers as good pedagogical tools, lack of training, and the time required to understand and use computer tools (DiSessa, 2000; Hannafin and Hill, 2002). This explains why some teachers shy away from computers and authoring applications with unintuitive interfaces, despite the possible quality of the outcome. A key remark that can be extracted from these works is the importance of the time it takes to accomplish a task with a technology (or rather the perceived time) in regard to the general willingness to adopt it as a pedagogical tool by teachers.

Obtaining an optimal flow during the learning process of a technological tool requires consideration of the variety of users and their different needs and motivations. Many factors influence how people think and reason, their knowledge, and their skills. Consequently, when designing a user-centered system, it is crucial to account for these variations (Barfield, 1986). In the end, this is a problem of diversity.

Users can be classified into four categories according to Hackos and Redish (1998): i) novice users with little or no knowledge of a tool, ii) regular users with basic knowledge of a tool, iii) advanced users who can perform some technical tasks, and iv) expert users or professionals with deep knowledge of the tool. Ideally, users would be able to increase their knowledge over time and move to higher categories. However, several fundamental barriers impede this progression. First, as mentioned, excessively complicated basic use of the tool can lead to early abandonment by novice and regular users, in the same way that a limited functionality may drive away more experienced users. Second, since not every user has the same needs, users may decide to remain in a particular category that is complex enough to satisfy their needs and feel no need to learn more advanced functionalities. Interface design can tackle these issues by hiding functionality to increase simplicity but making it easy for users with more expertise to find these functions to maintain constructive power.

There are two immediate issues arising from the user evolution process. Firstly, users abandon the tool, mainly due to difficulty or limited functionality. Both issues should be considered during the design process. The approach of discoverable interfaces showing functionality as required is thus one of the most important aspects of our design process. Secondly, not every user ends up being proficient and many stall in a particular category. This is a result of user idiosyncrasy, motivation, and needs. Thus, if the authoring tool satisfies the user's expectations, there is no need for improvement in this respect. In other words, there is a unique critical threshold at which the cost of further learning is higher than the benefits it brings. Reusing expert knowledge and feedback is an additional tool to minimize the learning cost of less-skilled users who have reached their personal learning threshold. For instance, in a word processor example, advanced or regular users reuse macros created by proficient users. In summary, these issues arise from the diversity of users, and therefore, it is crucial to keep in mind that users approach tools in very different ways.

3.2. Searching for regularities in different types of educational activities

One challenge for maintaining an optimal flow of learning is finding a consistent metaphor that eases the reuse of knowledge when using the tool. We wanted to identify which types of educational activities are used most often to devise a metaphor that encompasses their common characteristics in them. First, we reviewed the state-of-the-art authoring tools (see Table 1) and classified the type of activities they support. We found that all of them support multiple choice activities except ARIES (a system for

building augmented-reality interactive learning environments). Pair-matching activities are the next most popular type of activity (provided by five tools), followed by “fill in the gap” activities. Although less common, other types of activities can be found, such as word ordering, puzzles, crosswords, and memory games.

In addition, we conducted a study to determine the most popular among the educational projects designed by professional teachers using JClic. The JClic website has educational projects for different levels ranging from early childhood to university, and we randomly selected a sample of 189. Among these, 84 early childhood projects were included from a total of 284, while 105 were primary school projects from a total of 928. We analyzed 2586 activities from early childhood projects and 3471 activities from primary school projects and identified 14 different types.

The size of each sample (n) was calculated as follows:

$$n = (N \cdot \sigma^2 \cdot Z^2) / [(N - 1) \cdot ME^2 + (\sigma^2 \cdot Z^2)]$$

where N is the population size, σ is the standard deviation, Z is the z-score associated with the desired confidence level, and ME is the margin of error. N was set as the total number of projects (284 and 928 for each case). We assumed a σ of 0.5 since it is unknown in both cases. Z was calculated as 1.96 from a confidence level of 95%, and we chose a margin of error of 9%.

Fig. 1 summarizes the activity distribution of the analyzed educational projects created by professional teachers. The most popular activity was pair matching (33% in early childhood projects and 26% in primary school projects), closely followed by multiple choice activities (26% and 21%, respectively).

In summary, it seems that teachers prefer information, exploration, multiple choice, and pair-matching activities, which cover 78% of early childhood projects activities and 66% of primary school project activities. We developed DEDOS-Editor with this information in mind, focusing first on the design of multiple choice and pair-matching activities and providing a consistent interaction metaphor when creating them. This will give teachers a tool that will allow them to design the most popular activities without too much effort.

4. Design principles of Dedos-Editor

Based on the considerations described in Sections 3.1 and 3.2, we decided to use a card-based metaphor to implement the design principles of direct manipulation and consistency. In DEDOS-Editor, each activity is inspired by a card-based game in which the teacher decides the card content and the goals to accomplish (see Fig. 2). Cards are virtual objects that resemble their physical counterparts and are used similarly by dragging and dropping. One immediate advantage is that novice teachers can use their real world-knowledge for guessing how to define an educational activity. For example, an activity's layout is configured by dragging cards from a toolbar and dropping them on the board. Continuous feedback from the actions is visually reflected on the screen to facilitate understanding of the creation process and its progress. These features increase the predictability of the next action and reduce the effort in progressing with the tool.

Both multiple choice and pair-matching activities are consistently defined in DEDOS-Editor using cards (tokens). For multiple choice, this is done by choosing which tokens are correct answers to the activity by dragging and dropping a target on a card, while in pair matching tokens are connected with arrows (which are also placed by dragging and dropping). These tokens can be placed in a zone, which acts as a container for all the tokens and targets that are dropped in it. We experimentally evaluated whether this metaphor is flexible enough to incorporate

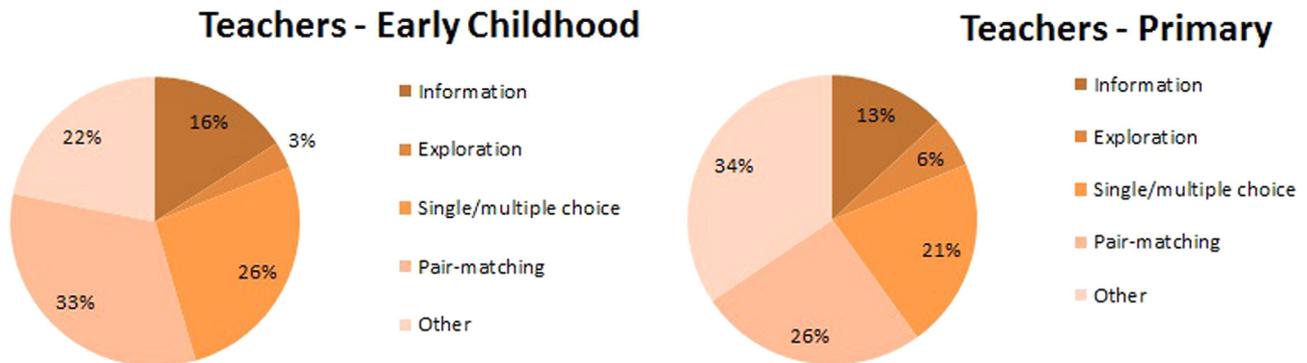


Fig. 1. Results of the most popular activities in ICT projects designed by current teachers. Other group is formed by 16 types of activities.

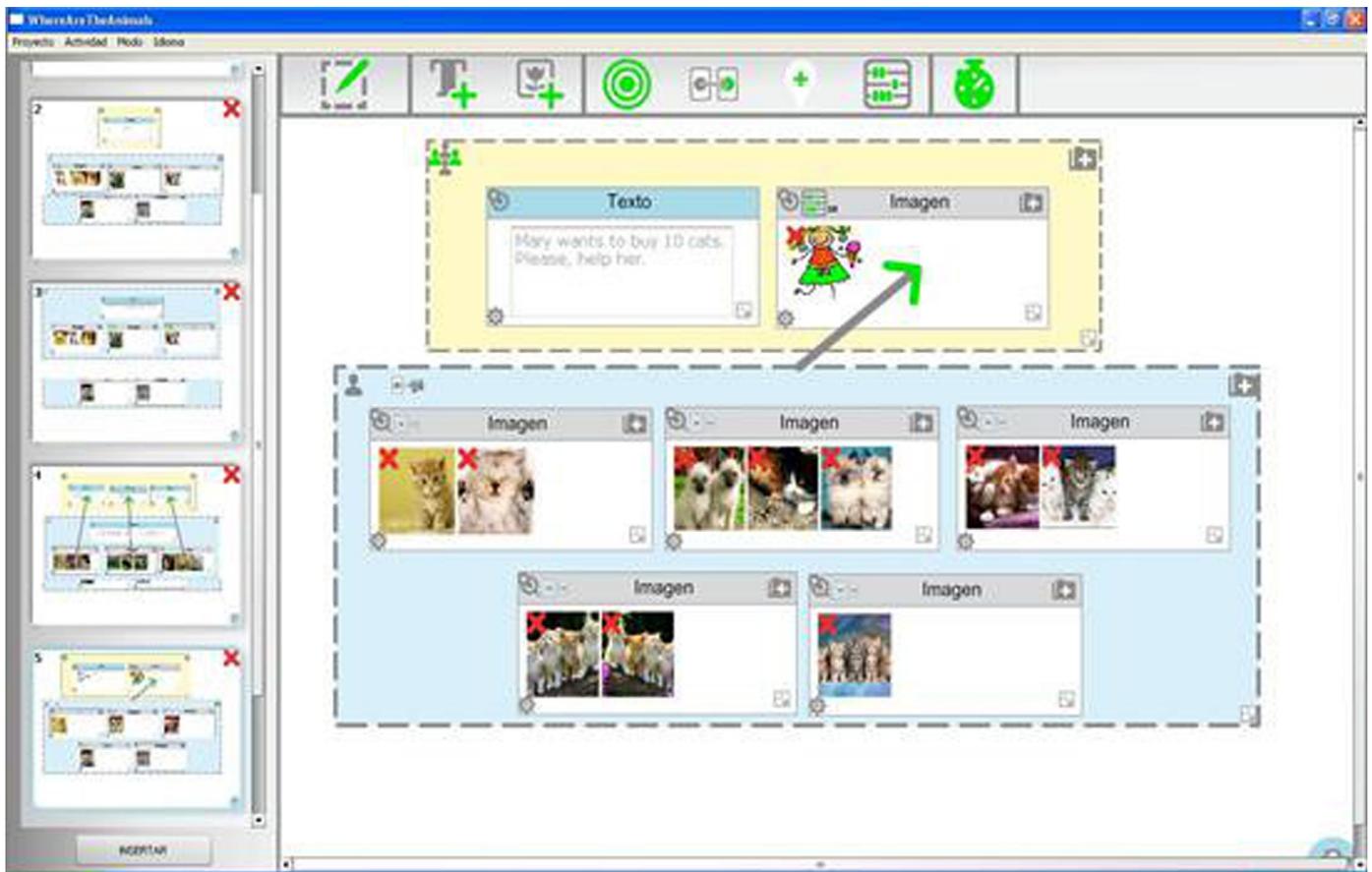


Fig. 2. DEDOS-Editor graphic user interface.

completely different activities in other tools, like word searches, crosswords, fill in the gap, memory activities, and so on. We believe that reusing concepts, elements, and operations provides an overall consistency that eases the creation of new types of activities by enabling reuse of the most needed concepts, as well as growing confidence toward mastery by eliminating conceptual jumps and strengthening a philosophy of “the more I know, the easier it will be to learn what I do not.”

We gauged whether DEDOS-Editor provides internal consistency in the creation metaphor. By reusing concepts, layouts, and actions in different types of activities, we intend to allow teachers to extend their goals easily to more complex tasks, which improves the learnability of the tool. In addition, we measured whether reuse improves the learning curve of DEDOS-Editor.

Taking into account user diversity (Hackos et al., 1998), the tool was designed to hide advanced configuration options from new users instead of showing the full potential of the tool right away. Advanced options are hidden on the back of the cards, which can be flipped to reveal the information. As teachers acquire experience with the use of the tool, they end up discovering these options to fine tune designs. This strategy avoids overloading new users and moderates the process of discovering new functionality and gaining competency. Logical constraints preventing novice users from making mistakes are also embedded in the user interface due to lack of knowledge. The next section explains the experimental study for evaluating the tool and learning paradigm in comparison with other tools based on menus. DEDOS-Editor is also described in more detail.

5. Validation

We experimentally measured the intuitiveness, progression, and retention of DEDOS-Editor (i.e., the ease of maintaining proficiency in the tool after a period of not using it). The experiment was also designed to compare with JClíc-Author. The experiment consisted of three sessions in which participants had to design different types of educational activities. Throughout all sessions, observers recorded both user issues and the time it took to finish activities. The next sections present the features of the participants and the experimental methodology.

5.1. Participants

The experimental study was conducted with 42 students aged 21 to 30 years old and majoring in either early childhood education or primary education at the Universidad Rey Juan Carlos. Participation was voluntary with no monetary compensation, but participants were awarded equivalent course points for the time spent. All students had previous knowledge of JClíc-Author as part of their basic training for making educational projects, but they all were using DEDOS-Editor for the first time. Since they had previous knowledge of JClíc-Author, we could expect good performance with the tool, so we could measure whether DEDOS-Editor is learnable enough to reach the same competence.

Participants were divided into two groups, one beginning the experimental study with JClíc-Author and ending with, while the other started with DEDOS-Editor. The participant distribution is described in Table 2 along with the number of participants in each group and the order of the tools used in each session. Groups were balanced to have a similar number of students. Degree type was not considered since both involve the same technological training.

5.2. Methodology

Each subject was asked to design different educational activities in three sessions. The first two sessions were done in one day, and students came back two days later to perform the third session. We separated the third session from the first and second since learnability is also measured by the knowledge retention. Information on how to use the tools was provided incrementally at the beginning of each session without giving any additional help for the rest of the sessions. After the three sessions, participants gave comments so we could gather their impressions on both tools. Details on the provided information can be found later subsections for each session.

Educational activities consisted of simple multiple choice or pair-matching activities. Participants had to design each activity in less than five minutes. If time ran out, participants could not finish the activity afterwards and they were told to start designing the next activity. We decided to use this time limit since it was about twice the time needed for the activities by the developers of this tool, who are experts in creating educational activities with DEDOS-Editor and JClíc-Author. Activities became complex throughout the sessions. All of the images that are required to build the activities were provided before starting the experiment. We did not want the participants to waste time looking for them since it would not affect the focus of our research.

Table 2
Distribution of participants during the experimental study.

	Participants	Initial tool	Final tool
Group 1	24	JClíc-Author	DEDOS-Editor
Group 2	18	DEDOS-Editor	JClíc-Author

Table 3
Distribution of the activities performed by session.

	Type of activity	Session 1	Session 2	Session 3
DEDOS-Editor	Multiple choice	2	2	2
	Pair-matching	2	2	2
JClíc-Author	Multiple choice	2	2	2
	Pair-matching	2	2	2
Information		No details	Explanation about the interface	Full explanation

Participants were first asked to design two multiple choice activities with one tool and then with the other tool, followed by two pair-matching activities. One group used DEDOS-Editor first and JClíc-Author next, which was reversed in the other group. Participants had five minutes to finish each activity before moving to the next one, and before designing each activity, we showed them how the final result should look. Table 3 shows an example of the activity distribution throughout the sessions and the information given to the participants.

The increasing difficulty made participants think about the spatial arrangement of the elements, since the clarity of the content could affect student learning. The activity content was the same for both tools. Although more types of activities are possible, we focused on only the two most popular for JClíc-Author. The previous experience with JClíc-Author helped strengthen the results should performance be better with DEDOS-Editor.

5.2.1. Session 1

No details were given about the tools in the first session other than where to find the main menus and elements. The first two activities of this session were multiple choice activities consisting of a text area stating the problem and three choices with only one correct answer. In the first activity, under the question, “Which of these animals lays eggs?” three graphical choices are to be shown using images of a gorilla, an elephant, and a turtle. In the second one, under the question, “Which of these countries does not belong to the European Union?” three text choices have to be provided for Canada, Germany, and Spain.

Using DEDOS-Editor in the first activity requires dragging and dropping one zone in the editing panel. The goal of the zones is to group cards. In this example, students must combine one game zone with one player zone. Individual zones will be replicated if there are two or more students doing the activity. However, collaborative zones are shared and located in the middle of the physical space. When the zones are added, students place a text token where they type the statement in the game zone and place the three image tokens for the answers in the player zone. Finally, users have to drag and drop a choice icon on the turtle token (the correct answer). One example of using collaborative and individual zones can be seen in Fig. 3. In the second multiple choice activity, the creation steps are the same as in the previous activity with the exception that text tokens have to be used instead of image tokens.

In JClíc-Author, multiple choice activities are called “identification activities.” The first step is to find and select this type of activity in an initial menu. Then, in the “Panel” tab, users can set the number of desired answers and the way they will appear (one row and three columns in this case) for creating an activity with similar look and feel to that in the other tool. This process creates a matrix that can be filled cell by cell by adding the appropriate

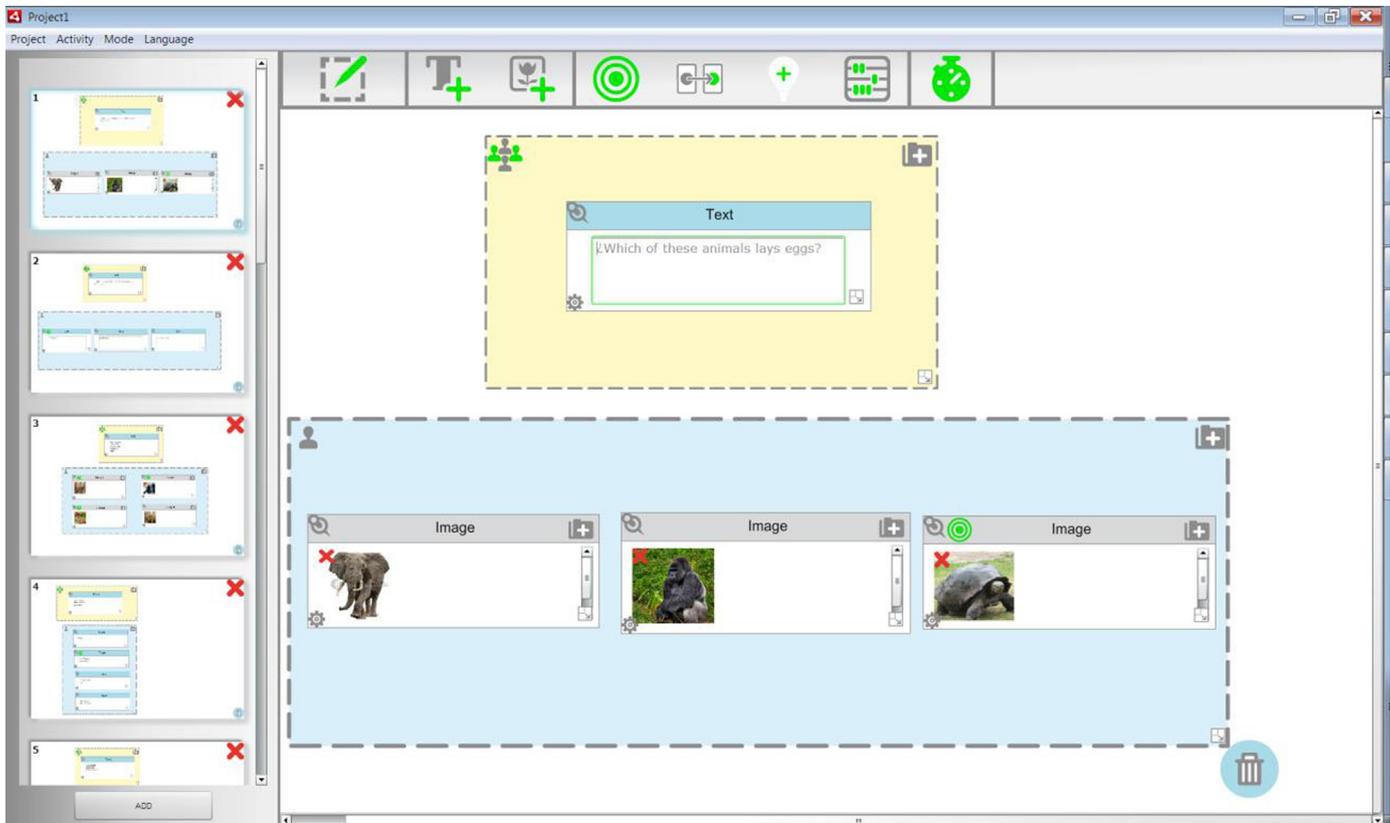


Fig. 3. Session 1. First multiple choice activity with DEDOS-Editor.

images and setting the correct answer in the “Relations” tab. Finally, the question statement has to be set in the “Messages” tab. An example of the result can be seen in Fig. 4. As with DEDOS-Editor, JClíc-Author allows the user to add text to the cells instead of images, which can be used for creating the second multiple choice activity.

The two pair-matching activities in the second part were an image-to-image activity to associate three types of products with a corresponding recycling bin and an image-to-text activity to associate three characters with their corresponding profession. In DEDOS-Editor, participants again have to drag a game zone and a player zone into the editing panel. In the game zone, they have to add the statement and three tokens that will be used as the destination of the tokens placed in the player zone. Participants should then place a pair-matching icon on each of the tokens and move the arrow that appears over the element intended for pairing. As with multiple choice activities, participants can use either text or image tokens interchangeably. Fig. 5 presents a recreation of one of the pair-matching activities designed with DEDOS-Editor where the relationships are visually represented by arrows between two cards.

In JClíc-Author, these types of activities are called “complex association,” and participants need to identify and find this type of activity in the initial menu to start the activity construction. Then, participants need to set up the number of elements of the relations in the “Panel” tab and choose the position of each cell as well as the images or text they should contain. JClíc-Author automatically creates a relation between each cell and the one above it, and which can be changed in the “Relations” tab. However, this process turns out to be non-trivial for many participants because the relationship between the different elements of the activity is not clear in the interface (see Fig. 6).

5.2.2. Session 2

Activities in this session were slightly more difficult and included more tokens in play and more multiple choice targets. On the other hand, participants were given more details about user interfaces. In the case of DEDOS-Editor, “zone,” “token,” and “target” concepts were explained, while for JClíc-Author, participants were shown how to create activities, use messages and panels, and set up targets in the “Relations” tab. After the explanations, we repeated the same methodology used in session 1 for designing two multiple choice and two pair-matching activities with both tools.

In the first multiple choice activity, we increased the numbers of tokens, images, and targets so that the user had to discover that several images can be added to the same element and that the same activity can have more than one correct answer. The second multiple choice activity required participants to use a different layout with a single column of options instead of the 2×2 matrix used in the previous one. As in session 1, these activities had to be done both with DEDOS-Editor and JClíc-Author. In the pair-matching activities, the number of tokens on screen was incrementally increased to force participants to manage the available space and adjust the zones and card size appropriately. There were also more elements than the first session in this case.

5.2.3. Session 3

Session 3 started with a full explanation of both tools, including how to take advantage of Windows Explorer thumbnails for easy location along with several detailed examples of solving the activities of previous sessions. This provided participants with full information to create activities in a timely manner and to make the most of both applications. Each participant had to design two multiple choice activities and pair-matching activities with both DEDOS-Editor and JClíc-Author. To analyze the skills acquired with both tools, these

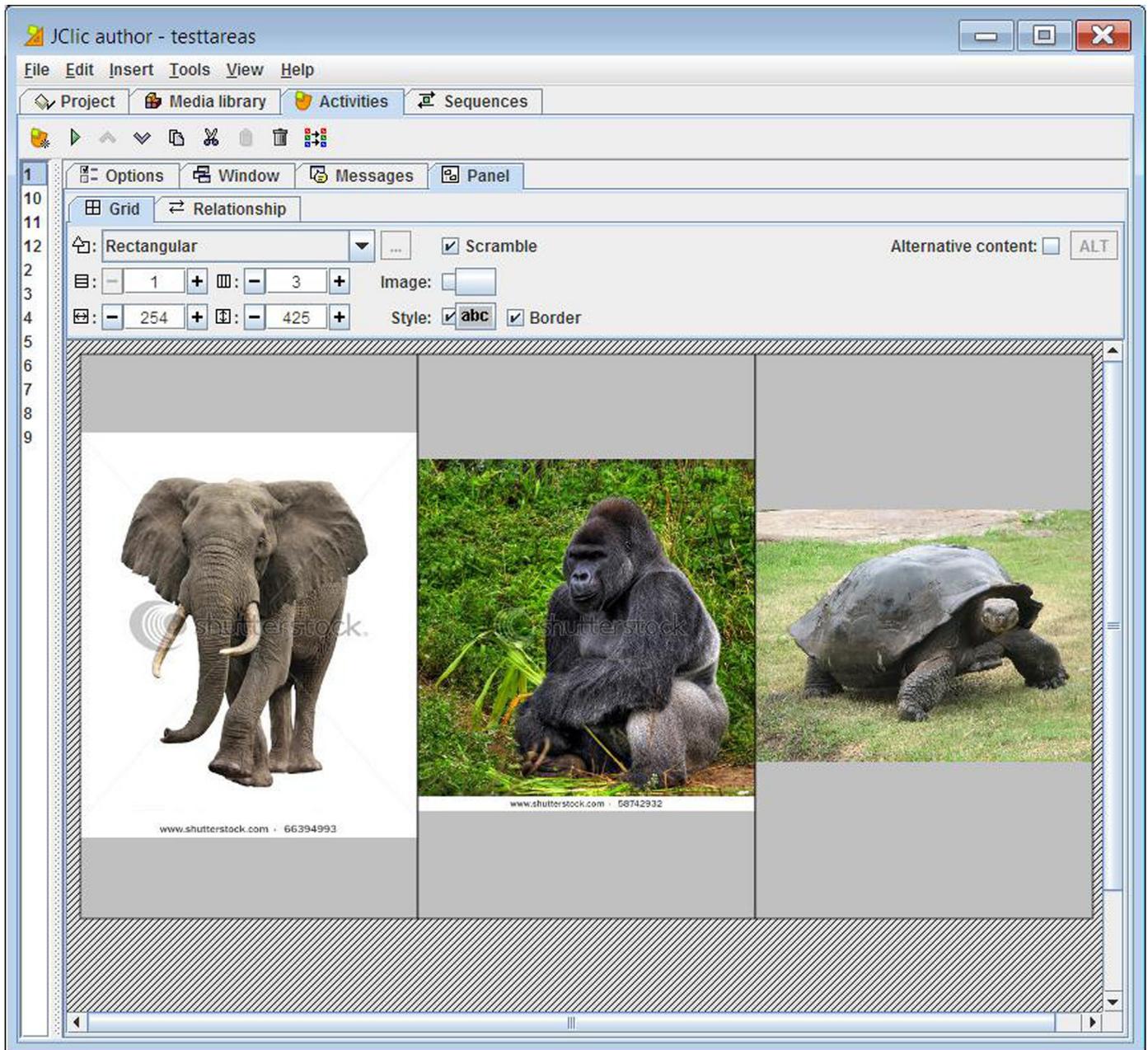


Fig. 4. First multiple choice activity with JCLic-Author.

activities included text and image tokens, multiple choice with one and with multiple goals, and a varied number of tokens in play.

5.3. Measures

One of the key points of the experimental study is the lack of information that forces users to experiment with the tool to discover its uses and possibilities. Gradually increasing the complexity also allows us to analyze how previously acquired knowledge helps the user in progressing to more complicated tasks. These factors allow us to evaluate the learning curve regarding the entry point, learning evolution, and competence acquired. We consider the entry point as the difficulty the user finds when using a tool for the first time with little or no knowledge, which is represented by the completion rate of the first activity of [session 1](#). The learning evolution is the rate at which users acquire skills

throughout use, which is represented by the slope of the completion rate curve as a user gradually masters tool activities. Finally, the acquired competence is the level of mastery a user can reach after fair use of the tool, which is represented by the last point of the curve of the completion rate from the last activity.

Since all these factors strongly depend on the experience and competence of the students in using computers and educational software, we analyzed the results by dividing participants in two groups according to their performance in the last session. We assume that the computer skills did not vary throughout the experimental study and thus consider the last session as a proper proxy of this competence. It is worth noting that data related to the user profile (e.g., course or age) did not provide any meaningful segmentation of the population.

We used the k-means method to make this division, where the objective is to divide N observations into X clusters in which each

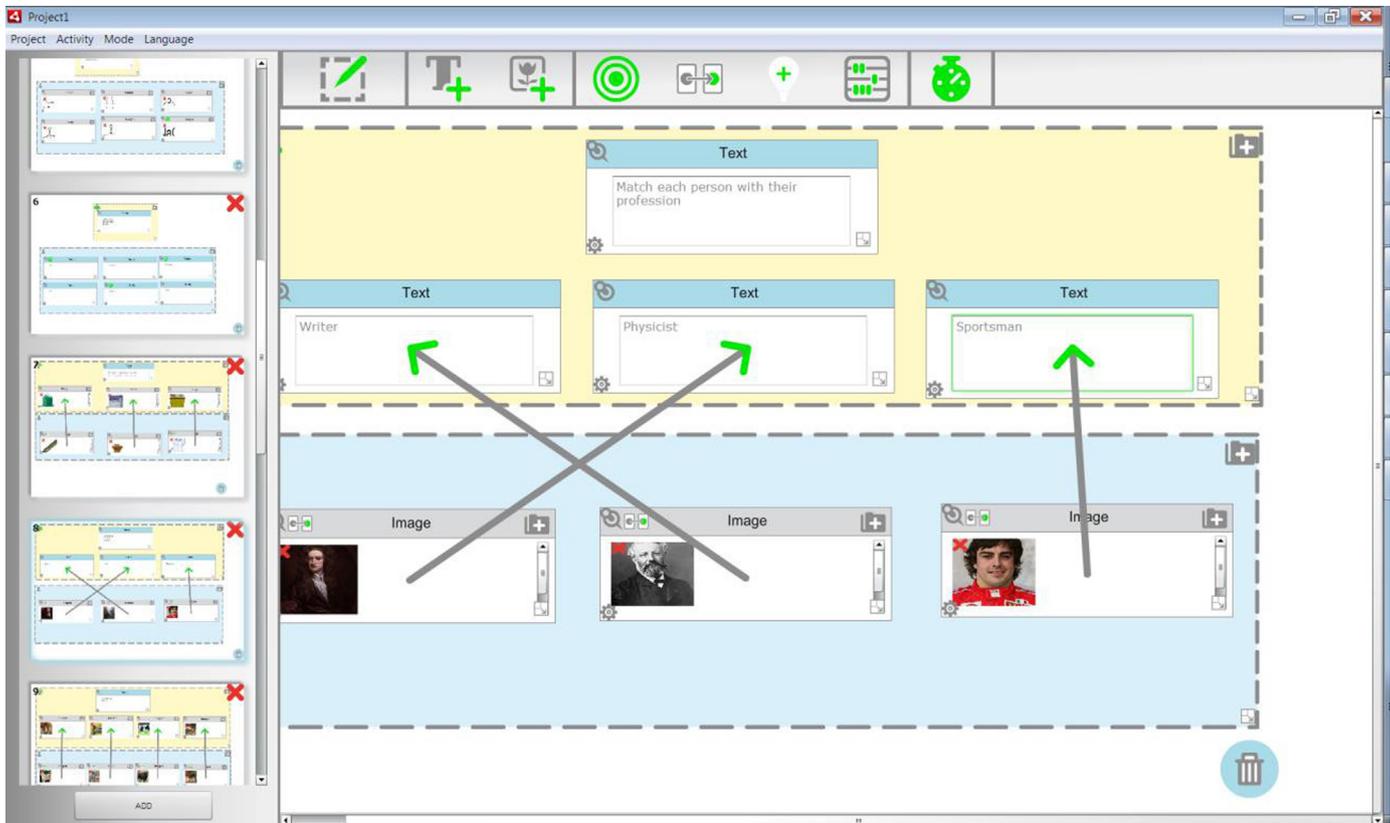


Fig. 5. Session 1. Example of pair-matching activity with DEDOS-Editor.

observation belongs to the cluster closest to its mean. With this method, users were divided into novices (15 participants, $M=165.5$ s) and experts (27 participants, $M=111.5$ s).¹ A chi-square test of independence was used to determine whether the order in which participants performed the experimental study affects the clustering (see Groups 1 and 2 in Table 2). The relation between these variables was not significant ($\chi^2(1, N=42)=0.08$, $p=0.78$; see Table 4). Thus, the participant allocation did not interfere with the students' performance. Using this segmentation, we analyzed the evolution of novice and expert users with both tools and compared their entry points, learning curves, and competencies acquired.

The next section presents the results obtained during the experimental study. We present the ratio of multiple choice and pair-matching activities during the three sessions grouped by type of users according to their technological skills.

6. Results

In analyzing the results of the sessions, we paid special attention to the completion rate of activities as a correlate of intuitiveness and ease of use by comparing the number of participants that finished each task for each tool in each session. We consider a task to be complete if the activity is designed correctly or the final result is equivalent to the requested one in less than five minutes. Sometimes, the same activity can be designed in different ways with both tools. We have used McNamara's test to compare the proportions of completion rates using each tool. This test checks the equality of proportions using the within-subject design of the

experimental study. In subsequent figures and tables, we use the abbreviation "Sel" to refer to multiple choice activities and the abbreviation "Emp" to refer to pair-matching activities.

6.1. Novices – multiple choice activities

As expected, we observed a steadily increase throughout the sessions completion rates of novice users in multiple choice activities, which resulted from gaining familiarity through use. Although not statistically significant, the completion rate was higher with DEDOS-Editor than with JClick-Author. However, we found a discrepancy in the first multiple choice activity in the third session, where the completion rate was much lower. This activity was a multiple choice activity similar to those designed in the first and second sessions. We believe that it was caused by the poor retention of the JClick-Author interface because of the many menus that the user has to memorize to use the tool well. This factor made the participants underperform in this activity and produced a considerable drop in the completion rate.

The results can be seen in Fig. 7 and Table 5. This table shows the number of participants who finished the corresponding activities within the given time. In addition, we can distinguish which participants finished the activity with DEDOS-Editor, JClick-Author, both applications, or neither of the tools. For instance, the table shows that a total of 10 participants from the novice group could not finish the first multiple choice activity with either of the tools, one participant succeeded with only JClick-Author, and four participants succeeded using only DEDOS-Editor.

6.2. Experts – multiple choice activities

Expert users show a higher entry point than novices (37% and 25.9% in DEDOS-Editor compared to 26.4% and 6.7% in JClick-Author) followed by a stronger increase in the completion rate that

¹ M is the mean time to complete an activity (measured in seconds) for each group considering only the activities in the last session.

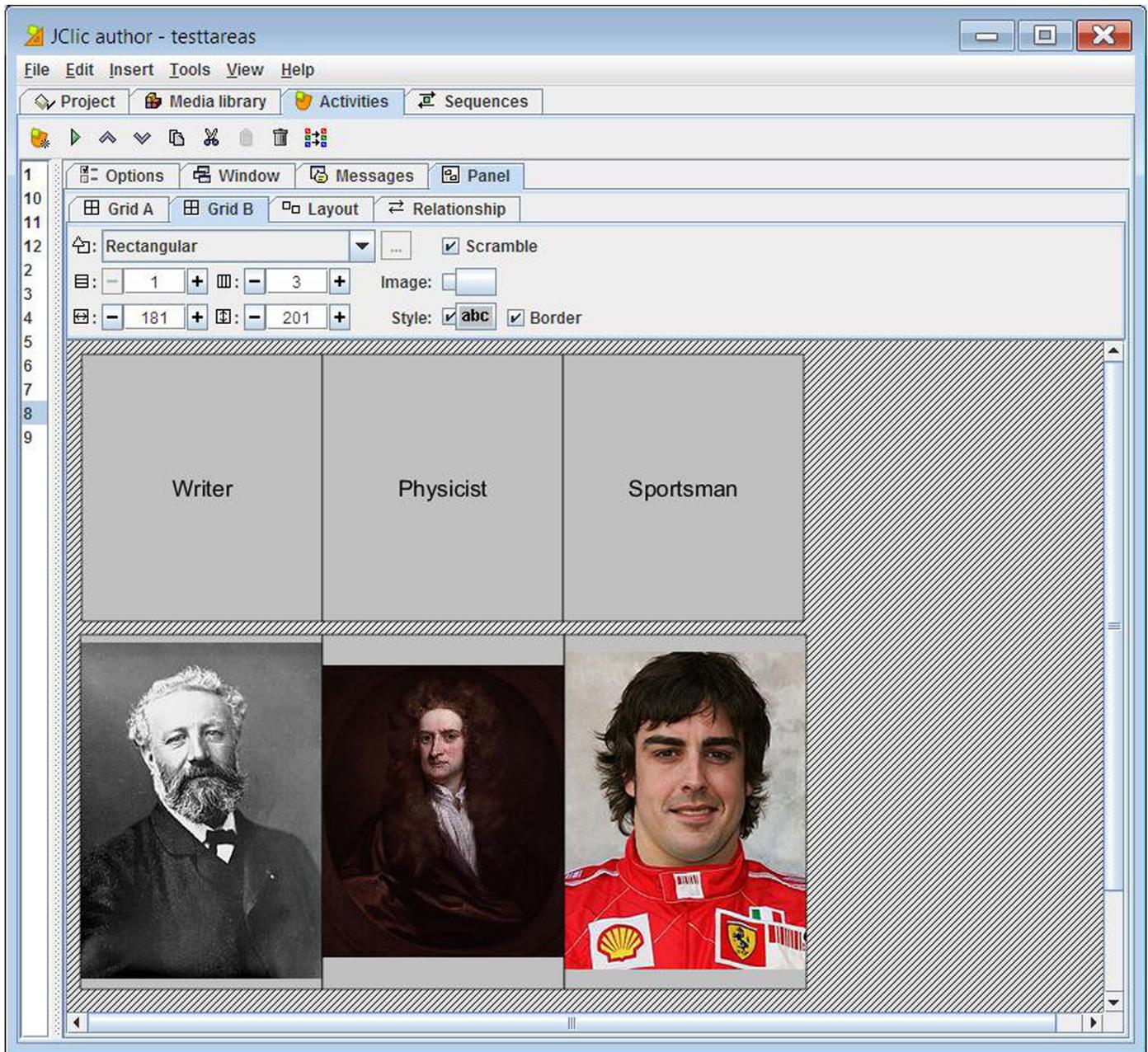


Fig. 6. Example of pair-matching activity with JCLic-Author from session 1.

Table 4
Distribution of the participants among the groups.

Performance clusters	Participant allocation		
	Group 1	Group 2	Total
Novices	9	6	15
Experts	15	12	27
Total	24	18	42

$\chi^2 = 0.08$ $p = 0.78$

remains over 75% for the rest of the experimental study with both tools. DEDOS-Editor seems to have higher completion rates in all activities except the second, but there is no statistical significance. Nevertheless, it is important to remember that DEDOS-Editor had not been used previously by any participant, whereas all participants had used JCLic-Author during their degree.

The activity with the smallest p-value was Sel3 ($p=0.025$),² in which users finished about 20% more activities with DEDOS-Editor than with JCLic-Author. This reaffirms our thoughts when analyzing the results of novice participants regarding retention for the JCLic-Author interface. Considering that users fall into the expert category, both learning curves are quite satisfactory with a reasonably high entry point, fast achievement of competence, and full competence acquired. The results can be seen in Fig. 8 and Table 6.

6.3. Novices – pair-matching activities

Pair-matching activities proved to be more difficult than the multiple choice activities. In addition, throughout the experimental study, we found that the amount of time spent by users for finding and

² After applying Bonferroni correction, the corrected p-value of Sel5 ($.025 \times 6 = 0.15$) is greater than the 5% significance level.

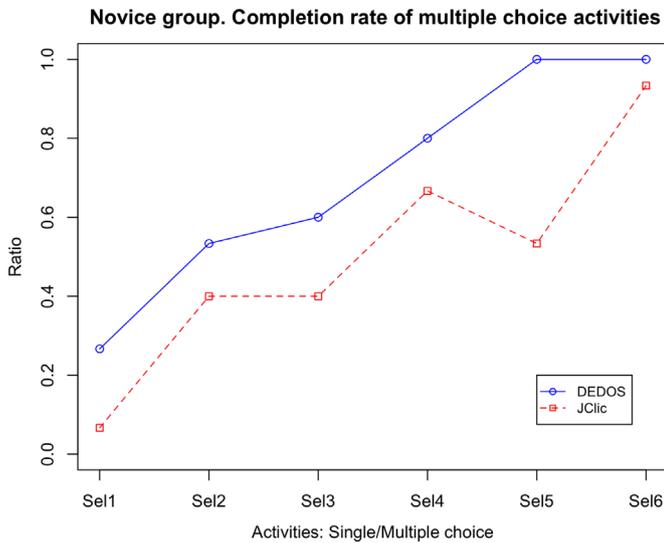


Fig. 7. Novice group. Completion rate of multiple choice activities.

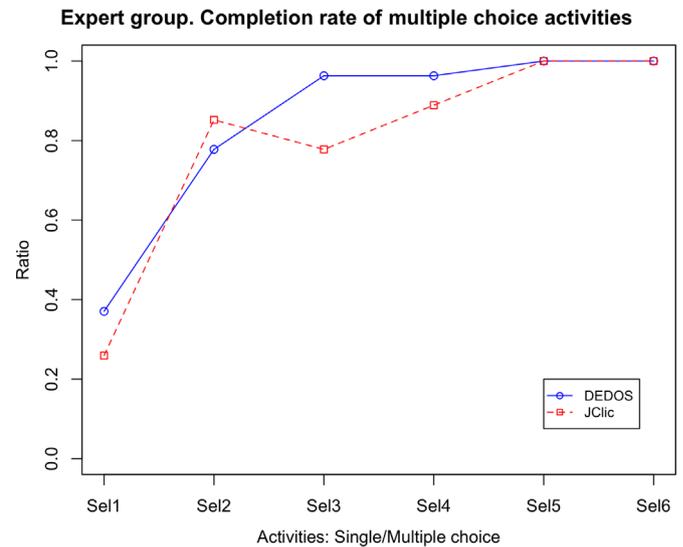


Fig. 8. Expert group. Completion rate of multiple choice activities.

adding images to the tokens or cells was significantly large compared to the total amount of time spent in finishing the tasks. Users had to insert from six to ten different images into the activity, and given the five-minute limitation, those who did not add the images fast enough tended to run out of time before completing the task.

The learning curves of DEDOS-Editor and JClcic-Author are significantly different in this case (see Fig. 9). DEDOS-Editor had a higher entry point than JClcic-Author, and this entry point is higher than that for novice users doing multiple choice activities using DEDOS-Editor. This can be explained by the fixed creation paradigm of DEDOS-Editor in that the experience users acquired in doing the multiple choice activities of session 1 served as an entry point for the pair-matching activities. In the second activity, the completion rate increased to 73.3%. In contrast, with JClcic-Author, less than 10% of users were able to finish the pair-matching activities of session 1 despite more than 50% having finished the multiple choice activities. This happened because designing multiple choice activities and pair-matching activities is fundamentally different in JClcic-Author, so the experience acquired in designing the former cannot really be exploited in the latter. In addition, finding the right template to begin with is not really intuitive, and participants had serious problems with choosing the correct type of activity from the initial menu. Astonishingly, this low rate did not improve until the end of session 2, despite users being explicitly told which type of activity they had to choose at

the beginning of the session. Overall, DEDOS-Editor shows a higher entry point followed by an immediate increase in completion rates, while JClcic-Author shows a flat curve that remained close to 0 until half the experimental study was completed.

In the second session with DEDOS-Editor, there was a slight decrease in the number of finished tasks. Since the first and the second sessions were performed on the same day, we believe this was caused by the large amount of images the participants had to add in session 2 rather than by users forgetting how to use the tool. The results are summarized in Table 7. It is worth noting that Emp8 and Emp9 are statistically significant, even after considering Bonferroni correction for multiple comparisons.

6.4. Experts – pair-matching activities

The analysis of expert users with pair-matching activities shows a remarkable difference in entry points (see Fig. 10). The completion rates differ by an order of magnitude in the first activity of session 1 (74.1% with DEDOS-Editor, 7.4% with JClcic-Author), which has special significance when considering that most participants had previous experience with JClcic-Author while none of them had any with DEDOS-Editor. As shown in Table 8, these significant differences between tools are visible during the first two sessions. Although applying the Bonferroni adjustment to Emp7, Emp8 and Emp10 makes them not statistically significant,

Table 5
Novice group. Completion rate of multiple (Sel) choice activities.

	Activity	DEDOS-Editor				X ²	p-value
		NO		YES			
		JClcic		JClcic			
Ses1	Sel1	66.67% (10)	6.67% (1)	26.67% (4)	0.00% (0)	1.8	.179
	Sel2	33.33% (5)	13.33% (2)	26.67% (4)	26.67% (4)	0.67	.414
Ses2	Sel3	33.33% (5)	6.67% (1)	26.67% (4)	33.33% (5)	1.8	.179
	Sel4	13.33% (2)	6.67% (1)	20.00% (3)	60.00% (9)	1	.317
Ses3	Sel5	0.00% (0)	0.00% (0)	46.67% (7)	53.33% (8)	7	.008**
	Sel6	0.00% (0)	0.00% (0)	6.67% (1)	93.33% (14)	1	.317

** indicates that the level of significance is below a certain value.

Table 6
Expert group. Completion rate of multiple choice (Sel) activities.

	Activity	DEDOS-Editor				X ²	p-value
		NO JClíc		YES JClíc			
		NO	YES	NO	YES		
Ses1	Sel1	51.85% (14)	11.11% (3)	22.22% (6)	14.81% (4)	1	.317
	Sel2	7.41% (2)	14.81% (4)	7.41% (2)	70.37% (19)	0.67	.414
Ses2	Sel3	3.70% (1)	0.00% (0)	18.52% (5)	77.78% (21)	5	.025
	Sel4	3.70% (1)	0.00% (0)	7.41% (2)	88.89% (24)	2	.157
Ses3	Sel5	0.00% (0)	0.00% (0)	0.00% (0)	100% (27)	–	–
	Sel6	0.00% (0)	0.00% (0)	0.00% (0)	100% (27)	–	–

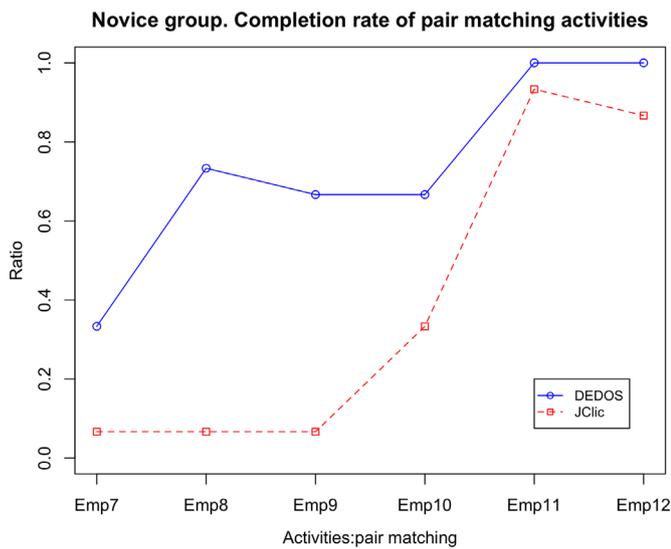


Fig. 9. Novice group. Completion rate of pair-matching activities.

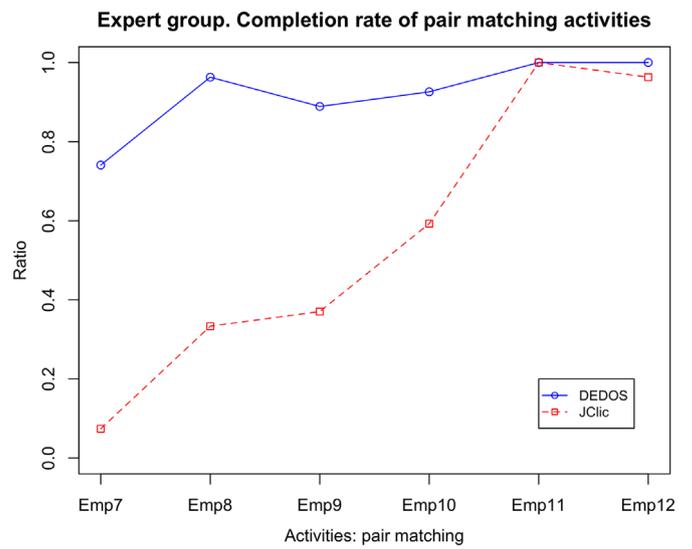


Fig. 10. Expert group. Completion rate of pair-matching (Emp) activities.

along with Emp9, they constitute more than half of the remaining tests showing a real effect.

In this case, it seems students need the full explanation for JClíc-Author to create pair-matching activities. Two factors can explain these results. First, the DEDOS-Editor user interface is based on a direct manipulation paradigm that avoids menus, right clicks, and other commonly used computer paradigms in favor of a

more direct and visual representation of options and actions. With this paradigm, intuitiveness plays a very important role in facing unknown situations so that solutions can be constructed by instinct without the proper knowledge or specific training. Second, the DEDOS-Editor user interface is designed as a holistic experience in which all possible designs emanate from common principles. There is no single choice that can constrain the possibilities of

Table 7
Novice group. Completion rate of pair-matching (Emp) activities.

	Activity	DEDOS-Editor				X ²	p-value
		NO JClíc		YES JClíc			
		NO	YES	NO	YES		
Ses1	Emp7	60.00% (9)	6.67% (1)	33.33% (5)	0.00% (0)	2.67	.102
	Emp8	26.67% (4)	0.00% (0)	66.67% (10)	6.67% (1)	10	.001**
Ses2	Emp9	33.33% (5)	0.00% (0)	60.00% (9)	6.67% (1)	9	.002**
	Emp10	26.67% (4)	6.67% (1)	40.00% (6)	26.67% (4)	3.57	.058
Ses3	Emp11	0.00% (0)	0.00% (0)	6.67% (1)	93.33% (14)	1	.317
	Emp12	0.00% (0)	0.00% (0)	13.33% (2)	86.67% (13)	2	.157

** indicates that the level of significance is below a certain value.

Table 8
Expert group. Completion rate of pair-matching (Emp) activities.

	Activity	DEDOS-Editor				X ²	p-value
		NO		YES			
		JClic		JClic			
		NO	YES	NO	YES		
Ses1	Emp7	25.93% (7)	0.00% (0)	66.67% (18)	7.41% (2)	18	.015*
	Emp8	3.7% (1)	0.00% (0)	62.96% (17)	33.33% (9)	17	.025*
Ses2	Emp9	11.11% (3)	0.00% (0)	51.85% (14)	37.04% (10)	14	<.001***
	Emp10	0.00% (0)	7.41% (2)	40.74% (11)	51.85% (14)	6.23	.012*
Ses3	Emp11	0.00% (0)	0.00% (0)	0.00% (0)	100% (27)	–	–
	Emp12	0.00% (0)	0.00% (0)	3.7% (1)	96.3% (26)	1	0.317

*. *** indicate that the level of significance is below a certain value.

a future design, and every new type of design needs only a small amount of new concepts. This allows users to take advantage of a variety of common concepts used in every possible design. Therefore, completion rates were similar between the first instances of both designing pair-matching activities and multiple choice activities with JClic-Author. In contrast, DEDOS-Editor users showed similar completion rates to the last time they designed multiple choice activities since they could use all the knowledge and experience gained in designing one type of activity when facing a new type of challenge.

Besides the difference in entry points, users with both tools display a satisfactory growing learning curve. In the case of DEDOS-Editor, we observed a slight drop in completion rate at the beginning of session 2 due to the increased number of the images they had to add, but the rate returned to 100% once users mastered image insertion. In JClic-Author, completion rates do not exceed 75% until the last session due to the extremely low entry point, despite the monotonic growth of the learning curves.

7. Discussion

DEDOS-Editor and JClic-Author were designed with the intention of providing teachers with powerful tools to improve the way

they transfer knowledge to the students. Both tools let teachers to design their own educational projects composed of learning activities that will be solved by students either individually or collaboratively. Teachers can design content for almost any topic and address all kinds of students. However, it seems that there are certain factors that prevent the full adoption of ICT by teachers. In this section, we discuss our experimental results to shed light on those factors that could affect adoption. Fig. 11 shows the completion rate of both multiple choice and pair-matching activities for all users in the order in which they were performed. Vertical lines delimit each session.

In general, the completion rates with DEDOS-Editor are always above those achieved with JClic-Author. In addition, both learning curves eventually show close to 100% completion rates through use and training, although entry points are below 40%. It is important that we did not provide any kind of mentoring with DEDOS-Editor at the beginning of the experimental study, even though it is highly advisable for improving teacher acceptance and knowledge when facing an authoring tool for the first time (Boulay and Fulford, 2009; Sampson, 2004). However, students should have full knowledge of JClic-Author and a positive attitude towards this tool since learning it is part of their degree. According to Kopcha (2012), participants should be more skilled with JClic-Author since they have had previous access to it and most of had

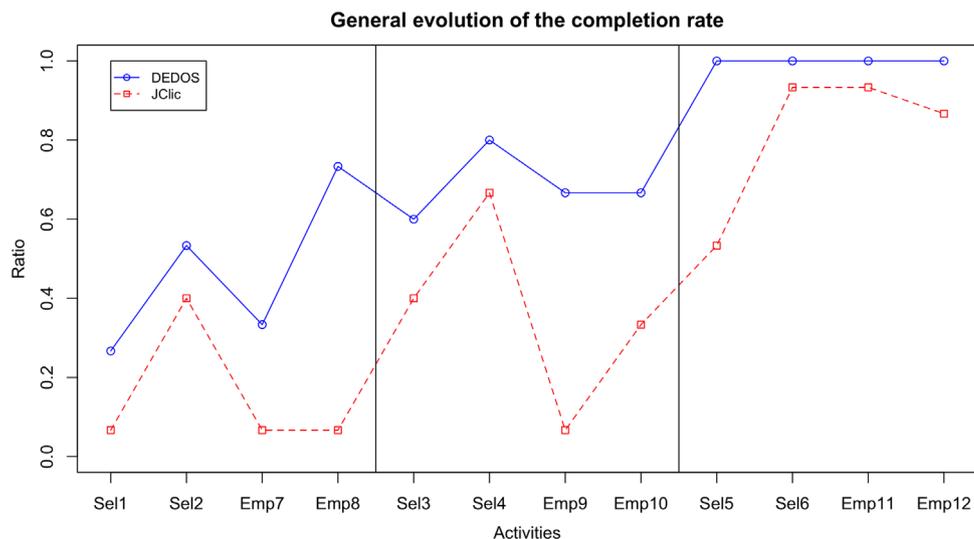


Fig. 11. General evolution of the completion rate.

used it in real classrooms during their internship placements. Nevertheless, our experimental data shows that having previous knowledge in a certain tool does not guarantee proper use of it in the future, which could prevent full adoption (De Grove et al., 2012; Aldunate and Nussbaum, 2013; Blackwell et al., 2013).

Although the entry points are quite similar with both tools, the difference in their learning curves is nonetheless impressive. DEDOS-Editor shows a smoother learning curve with small fluctuations when changing the type of activities, while JClc-Author displays drops when changing activity. As we have seen in previous sections, these drops do not occur when analyzing the learning curves of both types of activities separately. Nevertheless, due to the characteristics of JClc-Author, the knowledge gained when designing multiple choice activities cannot be used when designing pair-matching activities (“I do not even know where to start”³). Thus, when plotting both curves together (see Fig. 11), there are drastic drops corresponding to the learning curve intersections. As a holistic tool, knowledge and experience in one type of activity help boost performance in any other type in DEDOS-Editor (“Designing pair-matching activities with DEDOS seems way easier than with JClc”). This smoothens the learning curve as a whole, as shown in the upward trend throughout all sessions with DEDOS-Editor. We also believe that as the activities become more complicated to design (e.g., multiple choice activities are easier to design than pair-matching activities), the difference in design paradigms becomes more noticeable.

We estimated the complexity of both tools according to Murray’s variables (Murray, 2004) and discovered that the main problem of current authoring tools could be the great number of elements and functionality offered, which increases the difficulty when trying to manage all of them. In DEDOS-Editor, there are few elements that can be dropped into the editing area to create the activities, while JClc-Author has a more complex structure where elements have more parameters that must be learned to properly use the tool (“JClc has a lot of confusing options”). In contrast, our experiment showed that a little extra help allows users to completely master DEDOS-Editor in the last session, reaching 100% completion rate in all activities (“I have learned how to use DEDOS without asking anyone”).

When analyzing expert and novice users separately, we observe that not even expert users are capable of avoiding the drastic drops with JClc-Author (see Fig. 12). Advanced users with DEDOS-Editor, on the other hand, are able to master the tool at the end of session 1 and maintain a completion rate near 100% throughout the rest of the experimental study. In addition, users are able to reach completion rates similar to that of expert users with DEDOS-Editor at the beginning of session 3, while they fail to do so with JClc-Author. Therefore, background knowledge was not a determinant factor in the learning curve as other authors have stated (Ainsworth et al., 2004). If this were the case, the JClc-Author learning curve would be smoother than that of DEDOS-Editor since participants had previous knowledge of the tool. We believe that the complexity of the JClc-Author interface does not help the participants to retain the knowledge acquired from using the tool previously (Elliott et al., 2002), making it difficult when using the tool again after some time. However, further studies should be done with more participants since each of the groups (novices and experts) had different numbers of participants, which made the expert group more sensitive to differences.

DEDOS-Editor’s design to favor intuition and continuous learning is done through a direct manipulation paradigm and maintaining a strong core of common concepts regardless of activity type. We believe that these features help enhance the tool’s

generalizability. Despite not having the same amount of activities to design (only multiple choice, pair-matching, math, and point-connection activities), this paradigm choice has proven results in three significant outcomes, which have a close relationship with those presented by van Oostendorp and Walbeehm (1995) regarding direct manipulation interfaces. First, and most importantly, the competence acquired in designing one type of activity improves the general results of doing any other type. Second, with some experience and information, novice users rapidly gain expert competence in the tool. Finally, expert users (meaning those more fluent in the use of technology rather than having any prior experience with the tool) just need to experiment with the tool for a short to master it completely, even with a complete lack of information.

JClc-Author interface is based on menus and templates and offers a guided creation process in which the interface changes (more or less subtly) with user choices to offer only options relevant to the current development. Intuitively, this type of interface may seem a really good choice in which an apparently simple initial choice restricts options to those meaningful to that choice. However, the results suggest that the initial choice is not always as simple as intended by the interface developer, leading many users astray when dealing with an option for the first time. Second and most important is the clustered user interface were a tool has mini-tools for specific tasks drastically affects learning by clustering the learning process and majorly impeding reuse of knowledge in future tasks. To be fair, JClc-Author lets the user design a great variety of activities, so it is difficult to use the same design metaphor for all of them. However, this does not impede the developer in using the same metaphor in activities that are similar to each other (such multiple choice and pair-matching). Besides resulting in a slower learning curve, this issue gives rise to a dangerously frustrating situation in which users with relative mastery of the tool for one particular task see themselves pushed back to square one when dealing with another slightly different one.

Our experimental study was done with young students, so we cannot conclude whether the results would be the same if done with actual teachers. Nevertheless, participants did not have any previous knowledge about DEDOS-Editor and they gained expertise with this tool quickly throughout all the sessions. Thus, we could expect similar results with actual teachers, whose knowledge about DEDOS-Editor would be at least as high as the participants in this study.

Our research shows how different designs aimed to solve the same tasks (multiple choice activities and pair-matching activities) can produce a different user experience. The direct manipulation interface and the holistic, coherent creation paradigm seem to allow expert users to master the tool through just limited experience while allowing novice users to gain expert competence quickly with little information. Most importantly, the learning process allows a higher entry point due to the intuitiveness of direct manipulation and is smoother. In turn, the user take full advantage of any prior experience with the tool in dealing with new situations and thus avoid any frustrating drops in competence when facing new challenges. Although we need to perform more studies with different authoring tools, it seems that a direct manipulation interface and a coherent paradigm design help teachers to ease their workload, since they do not need to spend much time learning how to use the tool or creating the learning materials (Constantin et al., 2014). In addition, authoring tools with direct manipulation interfaces usually follow a “what you see is what you get” metaphor (e.g., Smart Notebook), which makes it easier to provide situational awareness to users and lets them clearly see the consequences of their actions. This makes the application easier to explore (Vigo et al., 2014) and therefore enhances the tool predictability.

³ Participant’s comment.

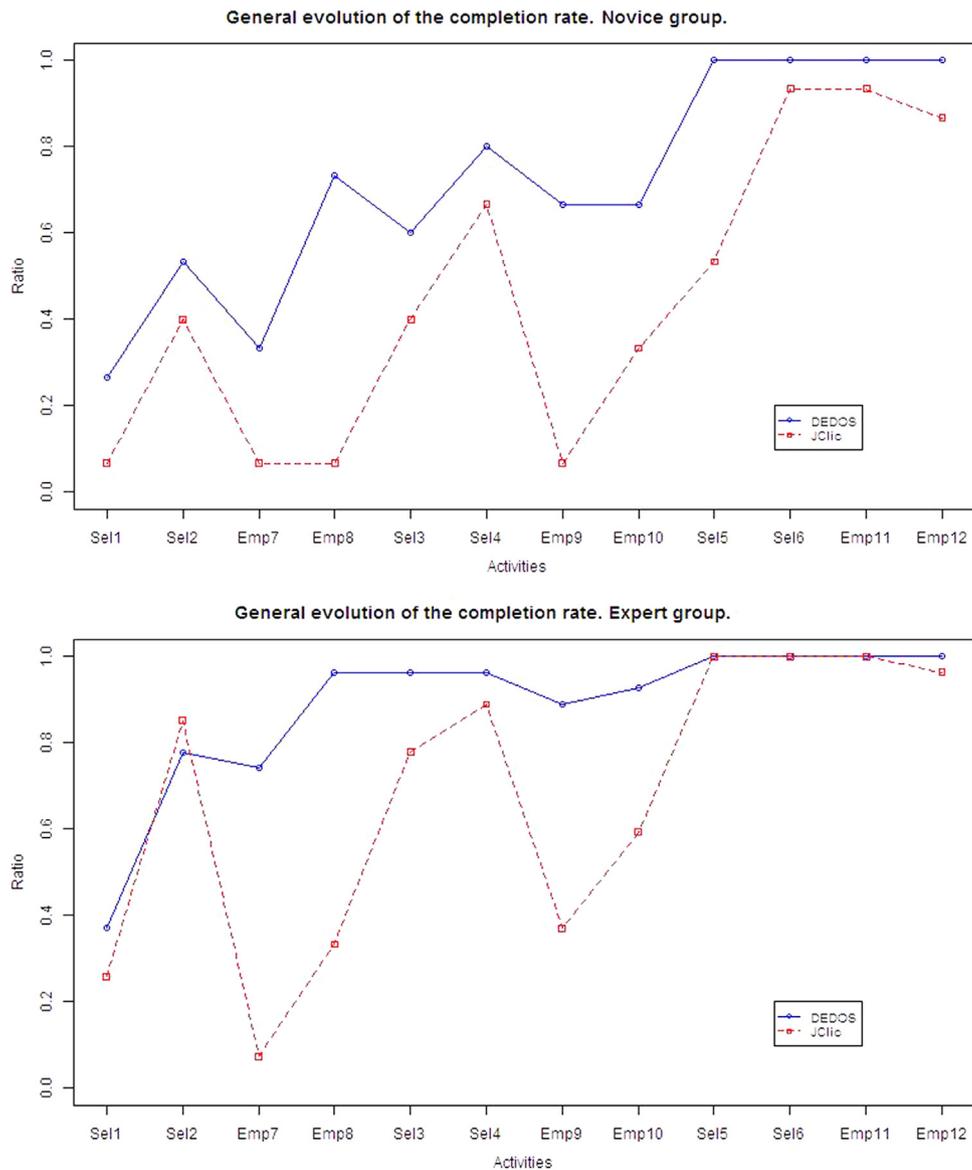


Fig. 12. General evolution of the completion rate divided by groups.

Authoring tools should focus on not only functional aspects but also on non-functional characteristics. Providing support to non-functional requirements is fundamental to allowing users of to complete their tasks (Lima et al., 2011). Most of the authoring tools reviewed seem to focus on only functional aspects and fail to provide efficient support to users. With our experimental study, we have made a first step towards clarifying which aspects could be relevant to teacher adoption of a certain authoring tool, showing that direct manipulation interfaces could help provide the non-functional factors that appear to be disregarded when designing new learning tools.

8. Conclusions

In the last decades, technology has been steadily entering the classroom in attempts to realize the greatest shift in education since the Industrial Revolution. Not only an information revolution but also a newer interaction revolution are claiming their space in education, pushing to redefine what and how students are taught. Nevertheless, while technology is undeniably entering the

classroom, the revolution is still waiting outside. As final deliverers of education, teachers have been plunged into a chaotic world of technologies and systems that are still far from their necessities and reach.

With this problem in mind, many researchers in technology-enhanced learning have made efforts to design and develop authoring tools to provide teachers with the necessary means to design their own content according to (ideally) their own teaching styles, necessities, and preferences. However, there is still a gap between the availability and the use of technology. More efforts are required to engage the teaching community in filling the void of content. While technology has received profound attention, and authoring tools exist to create for almost any application, we believe the failure of these tools is partially due to a low entry point when first used, as well as a slow and irregular progression during the learning process.

Considering the most preferred activities by teachers in authoring tool design is a key factor in promoting the adoption of ICT in education. However, further research should be done to increase the types of activities that can be created and improve the tools. This will let teachers create richer content that will help students in their learning processes.

We have aimed to shed light onto factors affecting the learning curve of an authoring tool and the final determinant of its adoption. While several tools are available, very few have been rigorously evaluated. We have presented an experimental study performed by future teachers (students of early childhood education and primary education degrees) and compared their interaction paradigms. A menu-based and inconsistent user interface design is the basis of JClíc-Author, the most used Spanish authoring tool, whereas a direct-manipulation interaction style and a consistent interaction metaphor were implemented in DEDOS-Editor, a novel authoring tool for traditional computers and multitouch surfaces. DEDOS-Editor turns teachers into developers, allowing them to create learning individual and collaborative activities or to change existing ones in an easy way. The experimental study showed that the user interface design choices have a tremendous impact on the learning curves of the tools. The direct-manipulation interface strongly affects the entry point, allowing completely new users to complete more tasks and gain knowledge through experience than with a menu-based user interface. In addition, the consistent interaction metaphor allows experience and expertise to persist through domains, resulting in smoother and faster learning curves. In contrast, choice-guided interfaces drastically decreased performance in slightly different tasks with the same tool, which could cause frustration and abandonment and should be avoided at all costs. The implications of user interface design in authoring tools for education require further study to fill the gap separating technology and education. We hope these results will offer valuable guidance for designers and serve as grounds for further studies in the scientific community.

Supplementary data

1. Videos:

- <http://www.youtube.com/watch?v=mgZjEP2zgCQ>
- <http://www.youtube.com/watch?v=aDdgm7pi1I>

1. DEDOS Youtube channel:

- <http://www.youtube.com/user/DedosMT?feature=watch>

2. DEDOS software - English version:

- <http://aprendecondedos.es/descargarte/aplicacion/>

1. Webpage of DEDOS Project:

- <http://aprendecondedos.es>. Most of the contents are in Spanish.

2. DEDOS Facebook page:

- <https://www.facebook.com/dedosmt>. This page is in Spanish.

Acknowledgments

This work has been funded by the Spanish Science and Innovation Ministry under project “e-Integra - e-Training y e-Coaching para la integración socio-laboral” (TIN2013-44586-R) and in part by the Region of Madrid under project “eMadrid - Investigación y Desarrollo de tecnologías para el e-learning en la Comunidad de Madrid” (S2013/ICE-2715).

Appendix A. Supplementary material

Supplementary data associated with this article can be found in the online version at <http://dx.doi.org/10.1016/j.ijhcs.2016.04.011>.

References

- Ainsworth, S., Grimshaw, S., 2004. Evaluating the REDEEM authoring tool: can teachers create effective learning environments? *Int. J. Artif. Intell. Educ.* 14 (3), 279–312.
- Aldunate, R., Nussbaum, M., 2013. Teacher adoption of technology. *Comput. Hum. Behav.* 29 (3), 519–524.
- Barfield, W., 1986. Expert-novice differences for software: implications for problem-solving and knowledge acquisition. *Behav. Inf. Technol.* 5 (1), 15–29.
- Blackwell, C.K., Lauricella, A.R., Wartella, E., Robb, M., Schomburg, R., 2013. Adoption and use of technology in early education: The interplay of extrinsic barriers and teacher attitudes. *Comput. Educ.* 69, 310–319.
- Boulay, R.A., Fulford, C.P., 2009. Technology mentoring: research Results across Seven campuses. *Int. Fed. Inf. Process.* 302, 273–281.
- Buabeng-Andoh, C., 2012. Factors influencing teachers' adoption and integration of information and communication technology into teaching: a review of the literature. *Int. J. Educ. Dev. Using Inf. Commun. Technol.* 8 (1), 136.
- Carroll, J., Howard, S., Peck, J., Murphy, J., 2003. From adoption to use: the process of appropriating a mobile phone. *Aust. J. Inf. Syst.* 10, 2.
- CAST Bookbuilder, 2006. (<http://www.nottingham.ac.uk/~cczjrt/Editor/>) (accessed 22.07.13).
- Clíc.xtec.cat. (2016). zonaClíc - JClíc. [online] Available at: <http://clíc.xtec.cat/es/jclíc/> [Accessed 22 July 2013].
- Constantin, A., Pain, H., Waller, A., September 2014. Using grounded theory methods to inform the design of an authoring tool. In: Proceedings of the 28th International BCS Human Computer Interaction Conference on HCI-Sand, Sea and Sky-Holiday HCI, pp. 177–182.
- Csikszentmihalyi, M., 1991. *Flow: The Psychology of Optimal Experience*. Harper Perennial, New York.
- De Grove, F., Bourgonjon, J., Van Looy, J., 2012. Digital games in the classroom? A contextual approach to teachers' adoption intention of digital games in formal education. *Comput. Hum. Behav.* 28 (6), 2023–2033.
- DiSessa, A., 2000. *Changing Minds: Computers, Learning, and Literacy*. MIT Press, Cambridge, MA.
- Dix, A., Finlay, J., Abowd, D.G., Beale, R., 2004. *Human-Computer Interaction*. Pearson Education Limited, Edinburgh.
- Elliott, G.J., Jones, E., Barker, P., 2002. A grounded theory approach to modelling learnability of hypermedia authoring tools. *Interact. With Comput.* 14 (5), 547–574.
- European Union (2013) Survey of Schools: ICT in Education. (<https://ec.europa.eu/digital-agenda/sites/digital-agenda/files/KK-31-13-401-EN-N.pdf>). (<http://dx.doi.org/10.2759/94499>) (accessed: 22.07.13).
- Hackos, J.A.T., Redish, J.C., 1998. *User and Task Analysis for Interface Design*, Wiley, pp. 76–87.
- Baked Half Software Inc., 2003. *Quandary*. (<http://www.halfbakedsoftware.com>) (accessed 22.07.13).
- Half-Baked Software Inc. and University of Victoria Humanities Computing and Media Centre, 2013. *Hot Potatoes*. (<http://hotpot.uvic.ca/>) (accessed 9.07.13).
- Hannafin, M.J., Hill, J.R., 2002. Epistemology and the design of learning environments. *Trends and Issues in Instructional Design and Technology*. Pearson Education, Inc, Upper Saddle River, NJ, pp. 70–82.
- Hutchful, D., Mathur, A., Joshi, A., Cutrell, E., 2010. Cloze: An Authoring tool for teachers with low computer proficiency. In: Proceedings of IEEE/ACM International Conference on Information and Communication Technologies and Development (ICTD'10). New York, NY, USA, Article 21. (<http://dx.doi.org/10.1145/2369220.2369239>).
- Karampiperis, P., & Sampson, D. G. (2004). A Flexible Authoring Tool Supporting Learning Activities. In: Proceedings of the CELDA, pp. 51–58.
- Kopcha, T.J., 2012. Teachers' perceptions of the barriers to technology integration and practices with technology under situated professional development. *Comput. Educ.* 59 (4), 1109–1121.
- LAMS International, 2004. (<http://www.lamsinternational.com>) (accessed 22.07.13).
- Levy, R.B.-B., Ben-Ari, M., 2007. We work so hard and they don't use it: acceptance of software tools by teachers. *SIGCSE Bull.* 39, 246–250.
- Lima, B.S., Soares, L.F.G., Moreno, M.F., March 2011. Considering non-functional aspects in the design of hypermedia authoring tools. In: Proceedings of the 2011 ACM Symposium on Applied Computing, pp. 1259–1266.
- Microsoft Learning, 2013. *Rapidly Create Online Courses*. (<http://www.microsoft.com/learning/en/in/training/lcds.aspx>) (accessed 9.07.13).
- Moraveji, N., Kim, T., Ge, J., Pawar, U.S., Mulcahy, K., Inkpen, K., 2008. Mischief: supporting remote teaching in developing regions. *ACM CHI08*, 353–362.
- Mueller, J., Wood, E., Willoughby, T., Ross, C., Specht, J., 2008. Identifying discriminating variables between teachers who fully integrate computers and teachers with limited integration. *Comput. Educ.* 51, 1523–1537.
- Murray, T., 2004. Design tradeoffs in usability and power for advanced educational software authoring tools. *Educ. Technol.-Saddle Brook Then Englewood Cliffs NJ* 44 (5), 10–16.
- Neto, A.T., Anacleto, J.C., Neris, V.A., Godoi, M.D.S., & Carvalho, A.F.P. (2006) A framework to support the design of learning objects based on the Cog-Learn pattern language. In: Proceedings of the 12th Brazilian symposium on Multimedia and the web, Natal, Rio Grande do Norte, Brazil: ACM. pp. 128–137.
- Nicholas, C. 2009. Personality-aware interfaces for learning applications. In: Proceedings of the 37th annual ACM SIGUCCS fall conference: communication and collaboration (SIGUCCS'09). New York, NY, USA, pp. 189–196. (<http://dx.doi.org/10.1145/1629501.1629536>).

- Nielsen, J., 1989. *Coordinating User Interfaces for Consistency*. Morgan Kaufmann Publishers, San Francisco.
- Papert, S., 1980. *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, New York.
- Pelgrum, W.J., 2001. Obstacles to the integration of ICT in education: results from a worldwide educational assessment. *Comput. Educ.* 37 (2), 163–178.
- Repenning, A., Ioannidou, A., 2006. What makes End-User Tick? 13 Design Guidelines. *Human-Computer Interaction Series* 9. Springer-Verlag, pp. 51–85.
- Shneiderman, B., 1983. Direct manipulation: a step beyond programming languages. *IEEE Comput.* 16 (8), 57–69.
- SMART) (2013) Smart Notebook. (<http://smarttech.com>) (accessed 22.07.13).
- Su, J.M., Tseng, S.S., Weng, J.F., Chen, K.T., Liu, Y.L., Tsai, Y.T. March 2005. An object based authoring tool for creating SCORM compliant course. In: *Proceedings of the 19th International Conference on Advanced Information Networking and Applications (IEEE)*. Vol. 1, pp. 209–214.
- Thomas, R.C., Milligan, C.D., 2004. Putting teachers in the Loop: tools for creating and customizing simulations. *J. Interact. Media Educ. (Designing and Developing for the Disciplines Special Issue* (15).
- van Oostendorp, H., Walbeehm, B.J., 1995. Towards modelling exploratory learning in the context of direct manipulation interfaces. *Interact. With Comput.* 7 (1), 3–24.
- Wojciechowski, R., Cellary, W., 2013. Evaluation of learners' attitude toward learning in ARIES augmented reality environments. *Comput. Educ.* . <http://dx.doi.org/10.1016/j.compedu.2013.02.014>
- University of Nottingham, 2008. XERTE. (<http://www.nottingham.ac.uk/~cczjrt/Editor/>) (accessed 22.07.13).
- Venugopal, S.T., Nor, N.F.M., Ayub, M.N., 2005. Development of multimedia authoring tool for educational material disseminations. *Inform. Educ. Int. J.* 4_1, 5–18.
- Vigo, M., Jay, C., & Stevens, R. (2014, April). Design insights for the next wave ontology authoring tools. In: *Proceedings of the 32nd annual ACM conference on Human Factors in Computing Systems*. pp. 1555–1558.