Computers in Human Behavior 54 (2016) 124–133

Contents lists available at ScienceDirect

Computers in Human Behavior

journal homepage: www.elsevier.com/locate/comphumbeh

Full length article

Reduced GUI for an interactive geometry software: Does it affect students' performance?

Simone S. Borges [a], Helena Macedo Reis [a], Leonardo B. Marques [a], Vinicius H.S. Durelli [b], Ig Ibert Bittencourt [c], Patrícia A. Jaques [d], Seiji Isotani [a, *]

[a] University of São Paulo, São Carlos, Brazil
[b] University of Groningen, Groningen, The Netherlands
[c] Federal University of Alagoas, Maceió, Brazil
[d] University of Vale do Rio Sinos, Rio Grande do Sul, Brazil

A R T I C L E I N F O

Article history:
Received 19 January 2015
Received in revised form
29 July 2015
Accepted 30 July 2015
Available online xxx

Keywords:
Interactive geometry software
iGeom
Graphical user interface
Experimental study
Interactive learning environment

A B S T R A C T

Purpose: The purpose of this paper is to describe an experimental study to reduce cognitive load and enhance usability for interactive geometry software.
Design/methodology/approach: The Graphical User Interface is the main mechanism of communication between user and system features. Educational software interfaces should provide useful features to assist learners without generate extra cognitive load. In this context, this research aims at analyzing a reduced and a complete interface of interactive geometry software, and verifies the educational benefits they provide. We investigated whether a reduced interface makes few cognitive demands of users in comparison to a complete interface. To this end, we designed the interfaces and carried out an experiment involving 69 undergraduate students.
Findings: The experimental results indicate that an interface that hides advanced and extraneous features helps novice users to perform slightly better than novice users using a complete interface. After receiving proper training, however, a complete interface makes users more productive than a reduced interface.
Originality/value: In educational software, successful user interface designs minimize the cognitive load on users; thereby users can direct their efforts to maximizing their understanding of the educational concepts being presented.

1. Introduction

The interplay between computer systems and human beings occurs through a graphical user interface (GUI). Improperly designed GUIs (i.e., interfaces that do not meet usability criteria) often hinder how users interact and access the underlying functionality (Nielsen, 1993). As a result, users might end up performing wrong operations, thereby reducing their productivity and even tampering with parts of the system. GUIs developed according to usability patterns provide a number of benefits to their users: they (i) optimize users' productivity; (ii) help users to quickly memorize the available functionalities; and (iii) mitigate interaction problems.

In the context of educational software for geometry, the development of interfaces can influence how learners explore and understand the concepts shown on the computer screen (Sedig & Liang, 2006). Interactive Geometry (IG) software is computer programs tailored toward geometry education. IG software allows learners to interact with geometry objects and dynamically construct their knowledge (Isotani & Brandão, 2008). Recent studies suggest that IG software containing interfaces that show a great number of graphical elements and features are not adequate for beginners (Kortenkamp & Dohrmann, 2010; Schimpf & Spannagel, 2011). According to these studies, when learners use GUIs containing a large number of functions, they spend a significant amount of time "trying to find certain features" instead of learning the subject (e.g. Mathematics).

* Corresponding author.
E-mail addresses: simone.borges@gmail.com (S.S. Borges), helenamcd@gmail.com (H.M. Reis), leobmarques@gmail.com (L.B. Marques), vinicius.durelli@gmail.com (V.H.S. Durelli), ig.ibert@gmail.com (I.I. Bittencourt), patricia.jaques@gmail.com (P.A. Jaques), sisotani@icmc.usp.br (S. Isotani).

http://dx.doi.org/10.1016/j.chb.2015.07.064

Although some studies have discussed the development of interfaces for IG software, little effort has been made to carry out empirical studies to evaluate how GUIs can be used to avoid problems in the learning processes (Mackrell, 2011). There are approaches to cope with the complexity found in GUIs. These approaches turn complex GUIs into more user-friendly ones. For instance, one approach consists of enhancing adaptability by allowing the user to either hide or disable extraneous features. By keeping only the graphical elements that are used more often, users tend to be more productive (Schimpf & Spannagel, 2011). However, the existing approaches are not specific to IG software (Kortenkamp & Dohrmann, 2010).

Designing effective GUIs for IG software is important because GUIs play a pivotal role in the learning process. Learning consists of transferring information from working memory to long-term memory (Coyne, Baldwin, Cole, Sibley, & Roberts, 2009). Nevertheless, our working memory can hold a limited amount of cognitive load, which means that when faced with difficult tasks we should use our working memory effectively. Within this context, learners that study geometry through IG software with low usability waste most of their cognitive load in learning how to interact with the underlying IG software. Thus, GUIs for IG software should be friendly enough not to exceed users' working memory capacity.

This study aims at understanding how the interfaces of IG software affect the learning process and the productivity of their users. In order to investigate this topic, an experiment involving 69 undergraduate students was carried out using the IG software called iGeom (Isotani & Brandão, 2008). Most subjects involved in the experiment stated they had advanced computer skills and a working knowledge of geometry. Only four students had prior experience with IG software, and only one student had prior experience with the particular IG software used in the experiment.

In the next sections, we present background, our experiment, and the conclusions of this study. The concepts of IG software, cognitive load, and usability are described in Section 2. Section 3 presents related work and similar experiments already carried out in the area. Section 4 describes the experiment we carried out using two versions of interface for iGeom, namely, **complete** and **reduced** interfaces. Section 4 also discusses the results of our experiment. Section 5 suggests future work and Section 6 presents concluding remarks.

2. Background

Interactive geometry software (IGS; dynamic geometry environments, DGEs; or dynamic geometry systems, DGSs) is computer programs developed with the goal of enabling students to explore geometry concepts through dynamic manipulation of geometric objects (e.g., Lines, circles and dots) (Erez & Yerushalmy, 2006; Isotani & Brandão, 2008; Roanes-Lozano, 2003). IG software implements the conventional tools employed in classroom settings to teach geometry, such as a ruler and compass using computational resources. The term "geometry" refers to the branch of mathematics that studies properties and relations of geometric objects. In this context, IG software allows students to create abstract representations of geometric objects and concepts to measure and manipulate them. These activities allow students to receive quick feedback after handling an object on screen. Consequently, students can test conjectures and hypotheses and find new relationships and properties based on a constructivist approach (Hollebrands, 2003).

During the learning process, students interacting with the software are able to visualize geometric constructions via GUI. Furthermore, they are able to interact with the features of the software and easily understand the information through these visualizations (Shimomura, Havannber, & Hafsteinsson, 2013). However, Baker, Greenberg, and Gutwin (2001) and Laborde (2007) suggest that the development of IG software should take into consideration not only pedagogical aspects, but also the design of the interface. The reason is to avoid developing software that tends to support a superficial sort of teaching, causing frustration among the students, who may struggle using the software and ultimately not direct their attention to the task that really matters: to learn geometry (Kortenkamp & Dohrmann, 2010; Schimpf & Spannagel, 2011).

Previous studies have shown that students who use IG software to learn geometry are more committed than students learning with traditional tools, such as rulers and compasses (Erbas & Yenmez, 2011). Research findings indicate that IG software encourages students to develop their own hypotheses and find new ways to solve the proposed problems (Isotani & Brandão, 2008). Yet, during the learning process students need to learn at the same time, both mathematical concepts and how to use the IG software (e.g., understand the interface and functions of the software).

2.1. Cognitive load theory and usability

Schimpf and Spannagel (2011) have shown that one of the main difficulties reported by students while learning how to use IG software is related to the wide variety of features in their interfaces. A large number of features can lead to ambiguity, confuse the students with too many details, and cause frustration and demotivate students. In other words, GUIs of IG software containing too many graphical elements may hinder the learning process, requiring users to experience high cognitive loads. As pointed out by Sedig and Liang (2006), the cognitive abilities of students are limited and should be directed to help them understand mathematical concepts and not be wasted learning how to use the software interface.

The cognitive load refers to the demands placed on the working memory of learners during the learning process. This concept is the basis of the cognitive load theory (CLT) (Paas, Renkl, & Sweller, 2003). According to this theory, humans have limited information-processing capacity. Because of that, we (humans) have difficulty memorizing past concepts for a long period during instruction. Cognitive load theory is concerned with how to maximize the performance of students investigating ways in which statements/interfaces must be presented and the type of activities in which learners must engage. The knowledge of human cognitive architecture provides the foundation that underlies the area, especially memory and long-term memory working.

CLT classifies cognitive load into three different types: intrinsic, germane and extraneous cognitive load (Sweller, Ayres, & Kalyuga, 2011). The ***intrinsic cognitive load*** is related to the complexity of a given task that must be processed by the learner and cannot be modified by an instructor. ***Germane cognitive load*** is the load utilized to the processing, construction and automation of schemas. The ***extraneous cognitive load*** is generated by the way in which information is shown to learners. It means that extraneous cognitive load demands mental efforts that do not promote learning and can be attributed to the design of the instructional materials (Chandler & Sweller, 1991).

The interface designers are concerned with the extraneous cognitive load, such as the level of irrelevant information presented to users. As shown in Fig. 1, if the interface presents irrelevant elements, learners need to figure out what information is important and what information is not relevant to their learning. Unlike intrinsic load, that is constant (considering the task to be performed); the other loads can be controlled (Gog & Paas, 2012). Therefore, to design an educational system that helps students to