Received: 15 December 2020 | Revised: 26 July 2021 | Accepted: 8 August 2021
DOI: 10.1111/jcal.12605

ARTICLE

Journal of Computer Assisted Learning WILEY

Using a distributed systems laboratory to facilitate students' cognitive, metacognitive and critical thinking strategy use

Joan Manuel Marquès Puig[1] | Thanasis Daradoumis[1,2] | Marta Arguedas[1] | Laura Calvet Liñan[1]

[1]Department of Computer Science, Multimedia and Telecommunications, Open University of Catalonia, Barcelona, Spain

[2]Department of Cultural Technology and Communication, University of Aegean, Mytilini, Greece

Correspondence
Thanasis Daradoumis, Department of Computer Science, Multimedia and Telecommunications, Open University of Catalonia, Rambla Poblenou 156, 08018 Barcelona, Spain.
Email: adaradoumis@uoc.edu

Abstract

Background: Recent research in online settings reports that supporting self-regulated learning (SRL) strategy use could lead to greater online academic success. A growing number of studies have started to investigate SRL supports in online environments recently, which indicates a great interest in this matter. Though several systems for automatic assessment of programming have been developed, there is hardly any study that has investigated how an automated assessment tool for distributed programming could facilitate students' SRL strategies.

Objectives: This study examined the ways our online Distributed Systems Laboratory (DSLab) tried to enhance students' SRL strategies in an authentic long-term online educational experience.

Methods: We applied an experimental research design, involving 111 university students who performed a programming assignment using DSLab. A customized questionnaire was used to collect data from all students.

Results and Conclusions: The statistical analyses revealed that DSLab tool managed to facilitate students' cognitive and meta-cognitive strategy use to a certain extent and critical thinking strategy use to a fairly large extent.

Implications: Though more experimental results are needed to delve more deeply into these findings, this study provides relevant implications for online distributed (or general) programming course teachers who seek to increase students' SRL strategies in this field.

KEYWORDS
cognitive, metacognitive and critical thinking strategies, online distributed programming learning, self-regulated learning strategies

1 | INTRODUCTION

Self-regulated learners are active participants in their learning, adopting various learning strategies, such as cognitive, metacognitive, critical thinking, and others, with the aim to manage their learning process better and improve their academic achievement (Panadero & Järvelä, 2015; Zimmerman, 2013). There are several self-regulated learning (SRL) strategies which students may employ to help them study, apply knowledge and skills and ultimately learn. These strategies fall into at least four categories: (a) cognition (strategies to remember or elaborate information); (b) metacognition (planning, setting goals, monitoring, and evaluating); (c) motivation (enhance self-efficacy, intrinsic task interest); and, (d) behaviour (help-seeking, time management, and creating a positive learning environment for learning task) (Andrade & Evans, 2013; Manso et al., 2016).

SRL strategies can be considered specific skills that can be exercised and reinforced when students use a specific learning environment to apply them in real contexts (Zimmerman, 2013). A systematic review performed by Wong et al. (2019) has shown that supporting SRL leads to enhance the use of SRL strategies and learning performance in computer-based learning environments. The importance of supporting SRL strategy use is further advocated by several meta-analyses which have evidenced a positive relationship between SRL-supports, SRL strategies, and academic achievement (Ergen & Kanadli, 2017; van Ewijk, 2011; Zheng, 2016). Another approach that supports SRL in online learning environments is feedback. Yet, in their literature review, Wong et al. (2019) found only two studies that investigated feedback as an approach to support SRL.

Though a variety of automated assessment tools has been developed over the years to assess programming assignments (Pettit et al., 2015), there is hardly any study that has investigated how an automated assessment tool for distributed programming could facilitate students' SRL strategies. Some studies in computer science education showed that students face difficulties when they are not aware of SRL and metacognitive strategy use; instead, when they employ SRL and metacognitive strategies they achieve good performance (Alharbi et al., 2012; Alhazbi & Hassan, 2013; Bergin et al., 2005).

In general, previous research has consistently shown that online learners need to self-regulate their learning. Embedding systems with SRL-enabling features in online learning environments is of increasing interest as one way to enhance students' SRL strategy use.

This study explores the extent to which our online Distributed Systems Laboratory (DSLab) can enhance students' SRL strategies when they deploy, execute and assess their distributed programming assignments using this environment. Feedback is one of the features that DSLab uses to support students in carrying out their assignment effectively. As such, one of the objectives of this study is to examine the effectiveness of the tool feedback in facilitating students' SRL strategy use.

The study will therefore address the following research question:

- RQ—To what extent has the DSLab tool managed to facilitate students' cognitive, metacognitive and critical thinking strategy use?

2 | LITERATURE REVIEW: SELF-REGULATED LEARNING (SRL) STRATEGY USE IN ONLINE COMPUTER SCIENCE EDUCATION

In their systematic literature review in computer science education, Garcia et al. (2018) highlight the value of e-learning tools that support students in developing SRL strategies and apply them to learn programming concepts and debug programs. Yet, their results showed that current tools are still far from achieving students' growth in self-regulated skills, opening up an opportunity for a more detailed investigation into these platforms to determine how students effectively engage in SRL. In their survey of automated assessment tools (AAT), Pettit et al. (2015) analysed the real usefulness of AATs in programming courses. They found out that AATs can be useful in reinforcing the accuracy of assessment, assisting teachers and enhancing student learning. Nevertheless, though several ATTs have been developed, only few studies were dedicated to producing formal results regarding the effects of tool use (Chen et al., 2017; Ihantola et al., 2010; Kelleher, 2014). In order to decrease negative student perceptions, the authors emphasize the importance for more systematic experimental research that takes the students' opinions into account for determining the necessary factors and features to improve AATs' design and use.

Other assessment tools highlight the use and importance of feedback when they analyse and assess programming assignments (Barker-Plummer et al., 2012; Stajduhar & Mausa, 2015; Vujošević-Janičić et al., 2013). All authors stressed the need to provide a feedback that is effective to the students, matching and coping adequately with the type and nature of student errors. However, all these systems do not provide a systematic way to explore the variety of attributes and factors that can promote the development of a programming assessment tool that matches the needs and interests of both teachers and students as regards its real usefulness.

In addition, few studies concerning automated programming evaluation systems attempted to examine ways for supporting students' SRL in these environments.

More specifically, some studies examined how to foster self-regulated learning in introductory computer programming courses (Alhazbi & Hassan, 2013; Bergin et al., 2005). To that end, several strategies, such as direct instructions, guided practice or feedback, were implemented using different tools to train student on SRL skills. The MSLQ instrument was used to measure students' awareness of SRL skills.

Alharbi et al. (2014) designed an online learning object system to support the self-regulated learning of programming languages concepts. They emphasized the importance of improving students' metacognitive skills by incorporating more features into the learning material, such as self-assessments with instant feedback, and self-reflection support.

Pedrosa et al. (2016, 2019) used the SimProgramming pedagogical approach to help students overcome learning difficulties transitioning from entry-level to advanced computer programming. To that end, the approach urged students to use as many SRL strategies as possible, namely organizing, planning and transforming strategies.

More recent web-based automated assessment tools for programming learning allow standard compiler/interpreter feedback and tracking of students' actions; however, the resulting analysis only provides information about student participation and progress (Gerdes et al., 2017; Grivokostopoulou et al., 2017; Hundt et al., 2017; Rivers & Koedinger, 2017; Robinson & Carroll, 2017;

Sukhoroslov, 2018). Neither study investigated the relationship of their web-based learning environment with students' SRL strategy use. Finally, no systematic studies exist that explore ways of supporting students' SRL in online environments used for distributed programming learning or for the automatic assessment of distributed programming assignments.

Given its importance for student learning, the aim of this work is to explore to what extent our automated programming assessment tool (DSLab) can enhance students' SRL strategies, focusing specifically in the area of distributed programming, a study which is missing in current literature.

3 | METHOD

3.1 | Context

DSLab is a web-based tool that provides a transparent deployment, execution and assessment of programming assignments (distributed algorithms) in a distributed infrastructure (a set of remote computers, each one running an instance of the algorithm). So, it is not a simple parsing tool but a stand-alone self-assessment environment. It differs from generic platforms like Learning and Course Management Systems (LMS and CMS), since the latter are used to manage and deliver e-learning courses and learning material. Yet, several SRL-supports were implemented in online learning environments, such as prompts, feedback, and other types of integrated support constructs (Wong et al., 2019).

Though general purpose learning platforms are not in principle designed to evaluate programs, from the development point of view it is feasible to extend them with an interface to manage the assessment of programs. DSLab could be integrated into a larger online learning environment, for instance through a generic API. However, the complexity of DSLab is not in the interface but in the other components that allow the assessment of programs in a distributed infrastructure. As a future work we may consider the case of providing a generic API to allow the integration of DSLab with different platforms, including a general purpose learning platform. But until now we have been focused on the development of the self-assessment tool with an ad-hoc interface with the aim to test its effectiveness on a variety of factors that affect students' learning of distributed programming.

DSLab allows teachers to define the evaluation tests that a student's programming assignment should go through so that it can be assessed as correct or not. More specifically, DSLab evaluates whether the distributed algorithm implemented by the student executes correctly. In an engineering degree it is very important that students demonstrate their ability to successfully accomplish practical assignments. Running manually all tests necessary to assess the correct functioning of a program is time consuming for teachers (some tests last up to 8 or 10 min), do not introduce any added value and it is error prone. On the contrary, using DSLab, students have immediate feedback and do not have to wait for the teacher's correction to know whether their code was correct of not. In addition to the assessment provided in DSLab, students deliver a report that includes the theoretical aspects of the assignment; in this case, teachers assess these reports manually.

The realization of our case study involved undergraduate students in a Distributed Systems online course who had to carry out a programming assignment. In fact, students had to implement the Time Stampled Anty-Entropy (TSAE) protocol, a distributed optimistic algorithm, which eventually guarantees that all replicas of a service have the same data. Since the implementation of a distributed algorithm is a complex task, we divided it in four parts (phases), as shown in Figure 1: Phase 1 involves the implementation of the functionality of the basic data structures necessary to implement TSAE; Phase 2 entails the implementation of a reduced version of TSAE protocol (only add operation, but no purge of log); Phase 3 requires the implementation of the functionality to purge log with unsynchronized clocks; and, Phase 4 concerns the implementation of a remove operation of a tiny service that uses TSAE protocol to maintain consistency between replicas. To that end, students could freely use the DSlab platform to upload each phase of their assignment and execute it in order to assess the correctness of its code. As soon as students achieved a correct code, they could pass to implement the next phase until they complete the assignment.

The benefits of online students employing DSLab tool are manifold (Marquès, Daradoumis, Calvet & Arguedas, 2020): (i) their solutions are assessed automatically (without teacher's intervention) in a consistent manner and they get immediate feedback from the tool, which enables them to know right away whether their code does not work adequately; (ii) they strive for a better solution and, therefore, better learning, by submitting their code multiple times (as shown in Figure 1) and improving it through the tool feedback; and (iii) they are provided with a grading facility, which lets them know the mark they obtain at the end of the execution.

Yet, the current version of DSLab presents some limitations which are important to be expressed for placing research findings in context: (i) the tool feedback is quite technical, so the generation of a more descriptive and explanatory feedback that is more adaptable to students' knowledge level is more desirable; (ii) feedback provides execution logs at a certain level of detail explained below, however a more rigorous analysis of the log files of students' interactions with the tool could be more helpful; and (iii) students need to spend some time in order to achieve high familiarization and constant activity with the tool.

Students receive two types of feedback as a result of a submission: (a) detailed information about the outcomes of the execution

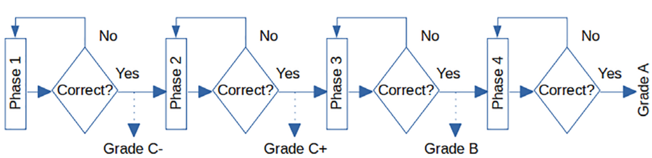

FIGURE 1 The four implementation phases of a programming assignment executed in DSLab

(this information depends on the assignment, therefore it is different for each type of assignment) and (b) execution logs (which we explain in detail below). This immediate feedback constitutes an important feature of DSLab. Figure 1 (result of all submissions) and Figure 2 (execution logs) illustrate how feedback functions in DSLab and is presented to a student.

Students can check the result of all their submissions. Figure 2 shows the result of the last six submissions of student s. "id" is a unique identifier of the submission. Other fields are:

- "Experiment": indicates the id of the uploaded code.
- "Result Date": date that the activity was submitted.
- "Group": id of the group that submitted the activity.
- "Success": indicates if the submission was correct or not.
- "Phase": id of the activity.
- "Result summary": summary of the result.
- "Result": link to the detailed information associated with the submission. In our case it is the final state of each data structure of each instance involved in the assignment together with the detailed information associated with the assessment.
- "Logs": link to a functionality that allows students to obtain the logs generated by each instance during the execution. Next (in Figure 3) we present an example of the feedback information included in the execution logs.

DSLab deploys and assesses the assignment (a distributed algorithm) in a distributed environment—that is, a set of computers each one running an instance of the algorithm. Students do not have access to these computers; therefore, they need a way to know what happened during the execution of the assignment, especially when the assignment failed. A common way of keeping track of the events that occurred while running a program is recording the information associated to these actions in a log. We implemented a logging service in DSLab to store the execution logs generated by each instance of the distributed algorithm while the assignment is being assessed. We defined six log levels (in decreasing order of severity): FATAL, ERROR, WARN, INFO, DEBUG, and TRACE. Students decide which logs to include in the code to trace the execution as well which log level to assign to each piece of information they want to log. This allows them to later filter the logs according to the level of detail they are interested in. Table 1 shows the six log levels, with their definitions and examples.

Figure 3 shows an extract of the execution logs generated during the assessment of the assignment. DSLab runs the student's implementation interacting with the instructor's implementation of the distributed algorithm. In this case, the assessment was done with eight instances of student's implementation interacting with seven instances of instructor's interaction, all of them running in different nodes in a distributed environment. The assessment is correct if the student's instances have the same final state as instructor's instances. The student can choose up to which level of detail he/she wants to get the logs; in this case, as we want to find out which the problem of student's code could be, we selected to view up to the level of TRACE, which is the level that provides more detailed information about what occurred during the execution of the assignment. More specifically, in this portion of the trace, we could see the following information: Instructor's instances 4 and 0 and Student's instance 6 are disconnected. Instructor's instance 2 and Student's instances 7, 0, 4, and 8 added a new date (a recipe in this case). Finally, at the level of TRACE, trace of instructor 2 at 08-04-2019 10:06:04:886 (*Inserting into Log the operation: AddOperation [...]*) provides a more detailed information about accessing the Log data structure. Similarly, the trace of instructor's instance 7 at 08-04-2019 10:06:06:753 includes the information that this instance is sending to a partner instance while being at a consistency session.

In general, students can look up the logs in the web interface (as shown in the snapshot) or download all execution logs in a zip file that contains a file for each instance. The file of each instance contains the logs generated by that instance.

3.2 | Participants

Participants were a sample of 132 fourth year undergraduate computer science students (93% male and 7% female), working adults with

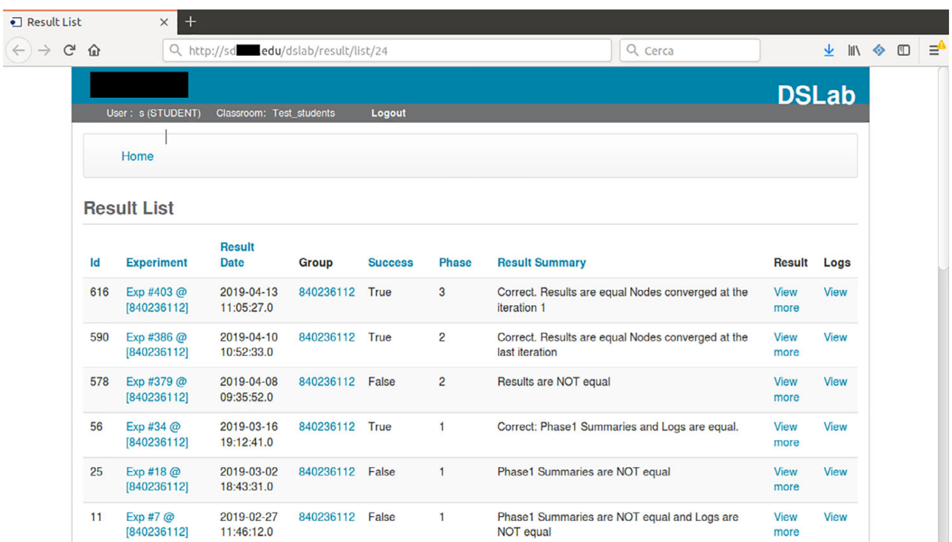

FIGURE 2 A completed assessment of an assignment in DSLab

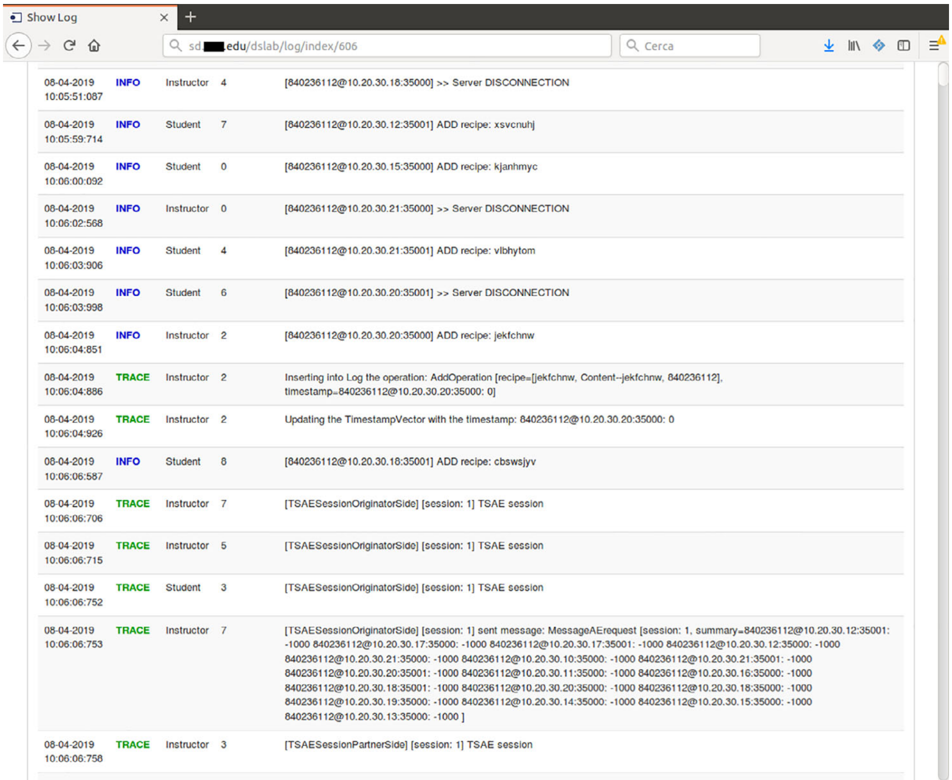

FIGURE 3 An extract of the execution logs of level INFO and TRACE generated during the execution of an assignment

TABLE 1 The six log levels of DSLab feedback with their definitions and examples

Log level	Description	Example	Explanation of the example
FATAL	Severe errors	Unable to connect to server 10.20.30.35	Server *10.20.30.35* is not reachable.
ERROR	Other errors or unexpected conditions	Not enough Servers were connected at the moment of finishing the Activity Simulation phase. Received Results: 8 numRequiredResults: 10	Unable to complete the evaluation because the number of available servers is lower than the minimum required number of available servers.
WARN	Undesirable situations that may not be an error	Recipe with timestamp [840,236,115@10.20.30.20:35000] is not inserted because other previous recipes are missing. Last known timestamp: [840,236,112@10.20.30.20:35000]	The data with timestamp *[840,236,115@10.20.30.20:35000]* could not be inserted because previous data is missing.
INFO	General information	[840,236,112@10.20.30.20:35000] ADD recipe: jekfchnw	A new data with timestamp *840,236,112@10.20.30.20:35000* is generated.
DEBUG	More detailed information	##### [iteration: 2/12] sending partial result	A partial result is send
TRACE	Very detailed information	Updating the TimestampVector with the timestamp: 840236112@10.20.30.10:35000: 2	The data structure TimestampVector is updated for the timestamp *840,236,112@10.20.30.20:35000*

a mean age of 36.68 years (SD = 16.58), who performed the same assignment during 7 weeks online. Students had mid- to high-level programming skills since they had previously attended three programming courses, an operating systems course as well as a computer networks and Internet course. The assignment could be carried out either individually or in pairs (formed by students themselves). In any case, students could decide which option they preferred since the same task and assessment criteria were applied.

3.3 | Research design

First, we present the theoretical basis that supports the proposed research and determines the variables we would like to measure. Panadero (2017) presented and analysed six models of SRL. He identified that Pintrich's SRL model and his widely used MSLQ questionnaire (Pintrich, 2004) constitute valuable instruments with highly significant impact in SRL research. Pintrich distinguished four different areas for regulation: cognition, motivation/affect, behaviour and context, which provides a comprehensive picture and a significant number of SRL processes. Due to the broadness of this construct and the novelty of our research, we focused our study on the cognition regulation area.

Regarding the area of regulation of cognition, Pintrich reported that there are a large number of cognitive strategies that students may use for memory, learning, reasoning, problem solving, and thinking. Indeed, many researchers have extensively studied the various *rehearsal*, *elaboration*, and *organizational* strategies that learners can use to control their cognition and learning (cf., Pintrich & De Groot, 1990; Schneider & Pressley, 1997; Weinstein & Mayer, 1986; Zimmerman & Martinez-Pons, 1986). *Rehearsal (repeating/memorizing)*, *elaboration (transforming)*, and *organizational (organizing)* strategies have been found to foster active cognitive engagement in learning and result in higher levels of achievement (Weinstein & Mayer, 1986). Indeed, recent research advocates that students who were trying to learn by *repeating/retrying*, *transforming*, and *organizing* learning material were more cognitively engaged and had better performance than students who tended not to use these strategies (Hwang et al., 2021; Lai et al., 2018; Zheng et al., 2020).

As for the metacognition scale, we considered items on metacognitive strategies that help students monitor their learning, namely *plan their learning* (e.g., set goals, or think of alternative solutions), *comprehension monitoring* (e.g., self-questioning before, during, and after reading to assist them to comprehend what they have been studying), and *regulate or change learning* (e.g., reread text to clarify or enhance meaning, or to get more clues to help them determine how to proceed next) (Isaacson & Fujita, 2006; Weinstein, Husman, & Dierking, 2000).

As concerns the critical thinking scale, we considered strategies that help students *apply prior knowledge to new situations* (e.g., be able to handle new related assignments with ease) and *analyse and evaluate information in a thoughtful manner* (e.g., analyse feedback and use it to recover from errors).

Based on the above, this study aimed to examine whether DSLab positively influences *Cognitive*, *Meta-cognitive and Critical* thinking strategy use (CMC). To that end, we defined the following variables: *rehearsal* (REH), *elaboration* (ELA), *organizational* (ORG) cognitive strategies, *metacognitive* (MET), and *critical thinking* (CT) strategies.

Students answered a questionnaire at the end of the learning experience. We designed a customized questionnaire with the aim to specifically respond to our RQ, which allowed us to collect quantitative data for our analysis. The questionnaire was based on and adapted from the Motivated Strategies for Learning Questionnaire (MSLQ) of Pintrich (2004), focusing on the cognition regulation area. MSLQ is a widely used instrument to assess college students' self-regulated learning (Chen, 2002) and its validity has been supported through extensive literature on college student learning and teaching (Gable, 1998; Taylor, 2012). The adaptation of the original MSLQ questions was guided by the particular context of our study, its application to the programming field and the features of our DSLab environment (e.g., the tool feedback).

3.4 | Data collection

The questionnaire was finally answered by 111 students, that is, more than 84% of the participants. The questionnaire aimed to examine *students' perceptions of cognitive, metacognitive, and critical thinking strategy use* (RQ). As such, it included 12 items associated with the RQ, as shown in Table 2.

In particular, we associated a set of questions to the five variables we defined for measuring students' cognitive, metacognitive and critical thinking strategy use. To that end we followed the reasoning and the established criteria of MSLQ in an accurate way, which makes our questionnaire a reliable instrument for our research. More specifically:

- **REH**: The use of *rehearsal* (*REH*) strategies is related to questions 11 and 12 (REH1-REH2)
- **ELA***: The use of *elaboration* (*ELA*) strategies is related to questions 2 and 4 (ELA1-ELA2)
- **ORG***: The use of *organizational* (*ORG*) strategies is related to questions 1, 3, and 8 (ORG1-ORG3)
- **MET***: The use of *metacognitive strategies* (*MET*) is related to questions 5, 7, and 10 (MET1-MET3)
- **CT***: The use of critical thinking strategies (*CT*) is related to questions 6 and 9 (CT1-CT2)

(the variables marked with * are explicitly associated with the tool feedback)

The use of specific metacognitive strategies (MET) has been captured by MSLQ question items which were adapted to examine whether students used ways for achieving *comprehension monitoring* (e.g., self-questioning—question 5), *regulation or modification of learning* (e.g., re-reading—question 7) and *planning of learning* (question 10—think of alternative solutions, that is, programming code variations, and test whether the tool provides flexibility by accepting them as correct). Similarly, the use of specific critical thinking strategies (CT) has been captured by MSLQ question items which were adapted to examine whether students used ways for *applying prior knowledge to new situations* (apply prior knowledge, obtained when students used the tool to execute previous assignment phases, to do a new phase—question 6), and *analysing and evaluating information in a thoughtful manner* (e.g., use prior experience to analyse information provided by tool feedback more thoughtfully—question 9).

A five-point Likert-type scale was used on a continuum from 1 (Strongly disagree) to 5 (Strongly agree); therefore, quantitative data is obtained.

3.5 | Data analysis

Data were analysed using descriptive statistics techniques, based on the Kolmogorov–Smirnov test for a sample and a frequency table. In addition, we calculated Pearson correlations between certain variables of our study with the aim to provide a more comprehensive answer of our research question.

The Cronbach's alpha coefficient was applied to the student data to ensure the reliability of data collection. In the analysis of questionnaire data, we obtained a Cronbach's alpha (0.78), which being higher than 0.70, strengthens the reliability of our items (Table 3).

In addition, we examined the coefficients of multivariate skewness and kurtosis for assessing multivariate normality. Critical values of all test statistics were calculated. The results showed that data were normally distributed, as absolute values of skewness and kurtosis did not exceed the allowed maximum (2.0 for univariate skewness and 7.0 for univariate kurtosis) (Table 4). The statistical results are presented in detail in the following section.

TABLE 2 Questionnaire of twelve (12) question items related to students' perceptions of cognitive, metacognitive and critical thinking strategy use (RQ)

	Students' perceptions of cognitive, metacognitive and critical thinking strategy use (RQ)	
1.	ORG1	When I worked on the assignment, I tried to put together the information from the course material, material from previous courses as well as information provided by DSLab tool.
2.	ELA1	When I worked on the assignment, I tried to analyse the tool feedback (logs) so I could correct the mistakes I have made.
3.	ORG2	It was easy for me to discern the useful information that the tool feedback (logs) provided to me.
4.	ELA2	When I was reading the tool feedback (logs), I was able to transform the hints I found into important ideas in my code.
5.	MET1	I asked myself questions to make sure I understood well the feedback provided by the tool.
6.	CT1	I used what I learnt from previous experience with the tool (in previous phases) to do a new phase.
7.	MET2	When I was reading the tool feedback I stopped once in a while and went over what I have read.
8.	ORG3	I outlined the best feedback I received from the tool so that it could help me do the next phases more effectively.
9.	CT2	When reading and analysing the tool feedback I tried to connect things with what I already knew.
10.	MET3	The use of DSLab made me the notion of correctness of the assignment too strict. (*R)
11.	REH1	I had to use the tool many times in order to achieve doing the assignment successfully.
12.	REH2	I chose to submit my assignment to the tool as many times as possible in order to guarantee its correctness (even though it was not necessary).

Note: Questions marked (*R) indicate those with negatively worded item (reverse item).

4 | RESULTS

We analysed students' perceptions of cognitive, metacognitive and critical thinking strategy use in relation to five parameters: (1) *REH*: Students' use of *rehearsal* strategies; (2) *ELA*: Students' use of *elaboration* strategies; (3) *ORG*: Students' use of *organizational* strategies; (4) *MET*: Students' use of *metacognitive* strategies; and (5) *CT*: Students' use of *critical thinking* strategies.

The corresponding descriptive statistic measures of the above items are presented in Tables 4 and 5.

Tables 4 and 5 indicate that the most significant values were obtained for variables REH, ELA1, ORG1, and CT. MET1 and MET2 variables were borderline significant. The rest of the variables (ELA2, ORG2, ORG3, and MET3) obtained non-significant values.

As regards *rehearsal strategies (REH)*, though many students (59%) agreed that they had to use the tool many times in order to achieve a successful assignment (REH1), a larger number of students (66%) consciously chose to submit their assignment to the tool as many times as possible in order to guarantee its correctness (even though it was not necessary). (REH2).

As concerns *elaboration strategies (ELA)*, most students (72%) were able to use the tool feedback to analyse the errors they have made in their code and correct them (ELA1). However, 37% were not able to transform the hints they found in the feedback into important ideas so that to achieve a more efficient or intelligent code (ELA2). Another 39% were not sure whether the feedback could provide them practical or useful clues for elaborating their code into an interesting program. Only 25% of the students managed to use this aspect of the elaboration cognitive strategy to obtain a higher level of programming code.

As for the *organizational strategies (ORG)*, a large number of students (65%) agreed that they were able to organize several pieces of information coming from different sources (such as course material, previous courses, DSLab feedback) in a meaningful manner so that to carry out the assignment. However, few students (19%) were able to dissociate the most useful information from the tool feedback so that to use it to improve their code. Moreover, just 30% of students tried to delineate the best feedback and use this information as a support for performing the next implementation phases more efficiently.

As concerns metacognitive strategies (MET), students mainly used them when coping with the tool feedback with the aim to monitor or change their learning. The two strategies students used indicate their persistence for using the tool feedback.

In particular, due to the importance of feedback for correcting students' coding errors, more than half of the students (54%) used the *comprehension monitoring* strategy (MET1) by asking themselves questions to make sure they understood well the feedback provided by

TABLE 3 Reliability statistics: The Cronbach's alpha coefficient

Cronbach's alpha	No. of elements
0.785	12

TABLE 4 Kolmogorov–Smirnov test for a sample

		REH1	REH2	ELA1	ELA2	ORG1	ORG2	ORG3	MET1	MET2	MET3	CT1	CT2
N		111	111	111	111	111	111	111	111	111	111	111	111
Normal parameters[a]	Mean	3.90	3.95	3.96	3.15	3.80	3.02	3.22	3.72	3.69	3.55	3.82	3.83
	SD	1095	1101	0.948	0.976	0.988	0.973	1099	0.912	0.940	1142	0.985	0.967
Test statistic		0.233	0.199	0.251	0.326	0.209	0.321	0.207	0.171	0.264	0.249	0.192	0.263
Asymptotic Sig. (bilateral)		0.000[b]	0.000[b]	0.000[b]	0.000[b]	0.000[b]	0.000[b]	0.000[b]	0.000[b]	0.000[b]	0.000[b]	0.000[b]	0.000[b]
		REH1	**REH2**	**ELA1**	**ELA2**	**ORG1**	**ORG2**	**ORG3**	**MET1**	**MET2**	**MET3**	**CT1**	**CT2**
Mean		3.90	3.95	3.96	3.15	3.80	3.02	3.22	3.72	3.69	3.55	3.82	3.83
Median		5.00	5.00	5.00	4.00	5.00	4.00	4.00	5.00	5.00	4.00	5.00	5.00
Mode		5	5	5	4	5	3	4	5	5	5	5	5
SD		1095	1101	0.948	0.976	0.988	0.973	1099	0.912	0.940	1142	0.985	0.967
Variance		1200	1213	0.899	0.953	0.976	0.946	1209	0.832	0.884	1304	0.971	0.936
Skewness		−0.442	−0.876	−0.839	−0.208	−0.972	0.042	−0.103	−0.829	−0.703	−0.382	−0.559	−0.799
Kurtosis		−0.628	0.373	0.355	−0.268	0.669	−0.269	−0.532	1392	0.973	−0.306	−0.076	0.372
Min		1	1	1	1	1	1	1	1	1	1	1	1
Max		5	5	5	5	5	5	5	5	5	5	5	5

Note: Grey values indicate the most significant values obtained in the analysis, whereas blue values are borderline significant.

[a]Test distribution is Normal.

[b]Lilliefors Significance Correction.

TABLE 5 Frequency data concerning students' perceptions of cognitive, metacognitive and critical thinking strategy use with regard to the five parameters: REH, ELA, ORG, MET, and CT

	REH1		REH2		ELA1		ELA2		ORG1		ORG2		ORG3		MET1		MET2		MET3		CT1		CT2	
	Fr.	%	Fr.	%	Fr.	%	Fr.	%	Fr.	%	Fr.	%	Fr.	%	Fr.	%	Fr.	%	Fr.	%	Fr.	%	Fr.	%
1	3	3	7	6	2	2	9	8	6	5	11	10	11	10	2	2	2	2	7	6	3	3	4	4
2	14	13	6	5	13	12	32	**29**	10	9	40	**36**	30	**27**	11	10	13	12	19	17	13	12	12	11
3	29	25	25	23	16	14	43	**39**	23	**21**	39	**35**	36	**33**	38	**34**	38	**34**	35	**31**	28	**25**	24	**22**
4	34	**31**	44	**40**	60	**54**	25	23	60	**54**	19	17	28	25	51	**46**	48	**43**	35	**32**	49	**44**	57	**51**
5	31	**28**	29	**26**	20	**18**	2	2	12	11	2	2	6	5	9	8	10	9	15	14	18	16	14	13

Note: Grey values indicate the most significant values obtained in the analysis.

the tool. Moreover, more than half of the students (52%) revisited those parts of the tool feedback that were not comprehensible to them. *Rereading* them more attentively helped students clarify problematic parts of their code and get clearer indications for changing their code and thus regulating learning (MET2). Yet, a significant number of students (46%) agreed that using DSLab to assess their assignment, the notion of correctness was too strict. Only 23% of them perceived that they could use DSLab to *plan their learning* differently (MET3), that is, allowing them flexibility by accepting different solutions—programming code variations—as correct.

As concerns critical thinking strategies (CT), we see that a majority of students used the two strategies examined. In particular, 60% of students *applied prior knowledge to new situations* (CT1), since they were able to use what they learnt from previous experience with the tool (in previous phases) to do a new phase. In addition, 64% of them were able to *analyse and evaluate information in a thoughtful manner* (CT2), when they were coping with the tool feedback. Indeed, when they were reading and analysing the tool feedback, they tried to connect things with what they already knew so that to get the most out of their endeavour to complete the assignment.

These results show that DSLab tool managed to facilitate students' cognitive and meta-cognitive strategy use to a certain extent. As concerns cognitive strategies, *rehearsal*, such as *retrying*, was a tool facility that was broadly used. Mainly, feedback has been the tool facility that was mostly associated with the other cognitive, meta-cognitive and critical thinking strategies. Feedback allowed *elaboration* since it incited most students to elaborate on the errors they have made in their code and correct them though it has not offered them more advanced elaboration strategies, like *transforming* (hints into important ideas in their code). Contrarily, feedback has not fostered students' *organizational* strategies use. These strategies were reflected mostly at a higher level since students were able to organize several pieces of information coming from different sources (such as course material, previous courses, DSLab feedback) in a meaningful manner. Metacognitive strategies (like "self-questioning" for *monitoring students' comprehension* or "skimming/rereading" for *regulating or changing their learning*) were employed to a considerable extent (more than 52%) when students were processing the tool feedback. *Planning students' learning*, like make them think of alternative solutions, was an issue that was not supported efficiently. Finally, critical thinking strategies, such as *apply prior knowledge to new situations*, or *analyse and evaluate information in a thoughtful manner* were supported to a fairly large extent (60% or more).

Furthermore, we provide the Pearson correlations between the cognitive variables *REH, ELA and ORG* and the metacognitive and critical thinking variables *MET* and *CT*, so that to identify whether *cognitive strategy use* has any significant influence on *metacognitive and critical thinking strategy use* (Table 6). We also provide the Pearson correlations between *MET* and *CT* variables (Table 7). We analyse these results in the Discussion section in order to obtain a more complete and knowledge-grounded response as concerns *the students' perceptions of cognitive, metacognitive, and critical thinking strategy use* when employing DSLab tool.

TABLE 6 Correlation between cognitive and metacognitive/critical thinking variables

	MET1	MET2	MET3	CT1	CT2
REH1	0.301**	0.178	0.161	0.142	0.202*
REH2	0.220*	0.303**	0.055	0.195*	0.264**
ELA1	0.314**	0.185	0.213*	0.434**	0.608**
ELA2	0.235*	0.182	0.198*	0.352**	0.472**
ORG1	0.277**	0.178	0.168	0.160	0.292**
ORG2	0.269**	0.161	0.180	0.256**	0.411**
ORG3	0.198*	0.171	0.260**	0.318**	0.537**

*Correlation is significant at the 0.05 level (2-tailed).
**Correlation is significant at the 0.01 level (2-tailed).

TABLE 7 Correlation between metacognitive and critical thinking variables

	CT1	CT2
MET1	0.209*	0.372**
MET2	0.122	0.229*
MET3	0.063	0.140

5 | DISCUSSION

In the field of programming learning assisted by automated assessment tools, DSLab provides an online environment that contributes to increase students' cognitive and meta-cognitive strategy use to a certain extent, whereas it facilitates critical thinking strategy use to a fairly large extent (RQ). This constitutes an important contribution to the field given that there are few attempts in the literature that examine ways for supporting students' SRL of programming languages in these environments. Yet, there is hardly any similar study in the particular field of distributed programming.

In general, the findings on students' cognitive, metacognitive and critical thinking strategy use in this study were also reflective of results in other studies. This supports the consistency of both the system and the results obtained. More specifically, working in the DSLab environment for executing and assessing their distributed programming assignments, students were allowed to use the *rehearsal* cognitive strategy by performing multiple submissions, which enabled them to use the feedback from previous executions to improve their code. This is consistent with the concept of iterative learning (Suleman, 2008) as well as with the contribution of this strategy to students' cognitive engagement and better performance (Zheng et al., 2020).

Similarly through its feedback, DSLab facilitated the use of the *elaboration* cognitive strategy. Feedback gave students the opportunity to analyse the errors they have made in their code and correct them. Providing elaborate feedback proves to be significantly related to the development of students' self-efficacy in web-based learning (Lai et al., 2018; Wang et al., 2013). In this regard, DSLab feedback should be further improved and enriched with prompts, such as self-explanation, reasoning-based prompts (Yeh et al., 2010), which could allow students to transform knowledge (Panadero, 2017). Such a construct could ameliorate students' use of the elaboration strategy with the aim to achieve a more efficient or intelligent code.

In addition, DSLab should certainly improve more the use of the *organizational* cognitive strategies. For most students it was easy to organize several pieces of information coming from different sources (such as course material, previous courses, DSLab feedback) in a meaningful manner so that to carry out the assignment (Bannert & Reimann, 2012; Pedrosa et al., 2016). However, DSLab feedback has not sufficiently contributed to foster the use of organizational strategies. These strategies can be a crucial SRL support to students, if they are capacitated to extract or outline the most useful information from the tool feedback so that to perform the programming assignment better (Biesinger & Crippen, 2010; Wäschle et al., 2014).

Regarding *metacognitive* strategy use, *comprehension monitoring* through self-questioning and *learning modification or regulation* through re-reading were two strategies that DSLab managed to facilitate to most students. Both items constitute important metacognitive strategies that help students monitor their learning (Isaacson & Fujita, 2006; Weinstein et al., 2000) as well as achieve good performance in computer programming (Alhazbi & Hassan, 2013; Bergin et al., 2005). Yet, a significant number of students found that DSLab notion of correctness was too strict. This fact impeded most students to *plan their learning* differently (e.g., think of and test alternative solutions), which makes the system being less flexible in accepting different programming code variations as correct. Despite this, by encouraging students' self-questioning and re-reading, DSLab feedback helped them be aware of SRL and metacognitive strategy use which, in turn, allowed them to achieve more efficient code. This finding is consistent with research which showed that students face difficulties in computer science education when they are not aware of SRL and metacognitive strategy use, so fostering them is important (Alharbi et al., 2012), whereas the use of feedback can be effective in supporting self-monitoring while performing problem-solving tasks (El Saadawi et al., 2010; Kramarski & Gutman, 2006; Lai et al., 2018).

Finally, DSLab makes an important contribution to the literature by providing an environment that clearly fosters critical thinking strategies, such as *apply prior knowledge to new situations*, or *analyse and evaluate information in a thoughtful manner* by means of the tool feedback. In contrast, the few studies that dealt with critical thinking strategies used other means such as critical thinking prompts in an implicit manner as reflection prompts (Ifenthaler, 2012) or as self-monitoring prompts combined with note-taking (Kauffman et al., 2011) to support self-monitoring and self-evaluation. Furthermore, social regulation strategies may be used to achieve more positive learning behaviours and indirectly enhance student higher order thinking (Hwang et al., 2021).

Most relevant to the above issues is the interplay between students' cognitive, metacognitive and critical thinking strategy use. That is, whether students' cognitive strategy use may have any correlation with their metacognitive and critical thinking strategy use. In addition, the relationship between metacognitive and critical thinking strategies is discussed. Both issues have not being previously discussed in the context of web-based automated assessment tools for programming learning.

Alexander et al. (2004) found that when students' progress toward higher levels of expertise and more complex tasks (such as the ones they have performed in our study), they need greater metacognitive monitoring and control. They also report that students who employ cognitive strategies more effectively, their activity will be connected with a related increase in metacognitive strategy use as well.

In the context of our DSLab tool, the Pearson correlations of Table 6 show that rehearsal strategies, like retrying, had a positive influence on metacognitive and critical thinking strategies that were related with the use of the tool feedback. That is, the students' repeated use of the tool and its feedback contributed to the use of specific metacognitive and critical thinking strategies.

Elaboration strategies that students used to analyse the tool feedback had a significant effect on critical thinking strategies and less influence on metacognitive strategies. This helped a large number of students (more than 60%) make the best of the tool feedback facility. Finally, students who employed *organizational* strategies were allowed to manage both their critical thinking and metacognitive strategies more efficiently.

The fact that students' cognitive strategies use had a rather positive correlation with their metacognitive and critical thinking strategy use is in line with previous research efforts which conclude that an effective use of cognitive strategies should be accompanied by a tendency toward metacognitive strategy use as well (Alexander et al., 2004; Dinsmore & Zoellner, 2018). Yet, few studies have analysed the relationship between cognitive strategies and metacognitive and critical thinking strategy use. For instance, the study of Kasimi (2012) revealed a significant correlation between students' use of cognitive and metacognitive reading strategies. Most studies examined this relationship toward other variables that this relationship affects, such as learning outcomes, learning styles, students' behaviour, etc. For instance, Saeedzadeh et al. (2018) showed that there is a significant relationship between cognitive and metacognitive strategy with the students' academic achievement.

Finally, previous research has explained the connection of metacognition and critical thinking and the ways metacognitive self-regulatory strategies were positively related to critical thinking (Garcia & Pintrich, 1992; Halpern, 1998; Ku & Ho, 2014; Kuhn & Dean, 2004; Magno, 2010). However, in our study, the Pearson correlations of Table 7 show that the metacognitive strategies that students used in the context of DSLab tool had no significant effect on students' critical thinking strategy use. The only significant correlation was the critical use of feedback information together with previous knowledge to carry out the assignment. Consequently, more effective metacognitive strategies are needed to increase students' metacognitive awareness and skills which, in turn, will enable higher improvement on students' critical thinking strategy use and skills (Çakici, 2018; Naimnule & Corebima, 2018).

6 | CONCLUSION AND LIMITATIONS

This work analysed the extent to which an online Distributed Systems Laboratory (DSLab) facilitated students' cognitive, metacognitive and critical thinking strategy use. Our results showed that when students upload, execute and assess their programming assignments in the distributed environment of DSLab, they perceive that:

1. DSLab allows them to use cognitive and meta-cognitive strategies to a certain extent and critical thinking strategy to a fairly large extent. In particular, feedback has been the tool facility that was mostly associated with the cognitive, metacognitive, and critical thinking strategies use. The *rehearsal* cognitive strategy was broadly used. The tool feedback prompted the use of *elaboration* strategies but has not fostered students' *organizational* strategies use. Here, a self-regulated inquiry approach (Lai et al., 2018) may be used to assist students in organizing information, coming from different sources (course material, previous courses, DSLab feedback, etc.), in a meaningful manner. Moreover, cognitive regulations applied in different groups can reduce tensions and enhance engagement, thus achieving effective collaborative learning (Zheng et al., 2020). Feedback also supported the use of *metacognitive* strategies (monitoring students' comprehension, and regulating or changing students' learning). In addition, *critical thinking* strategies (such as apply prior knowledge to new situations, or analyse and evaluate information in a thoughtful manner) were supported to a fairly large extent.
2. Regarding the interplay between students' cognitive, metacognitive, and critical thinking strategy use, in the context of our DSLab tool, cognitive strategies use had a positive correlation with students' metacognitive and critical thinking strategy use in many aspects. Instead, metacognitive strategies use had no significant effect on students' critical thinking. Here, since students may work in small groups to carry out their assignment, the social regulation-based online learning approach (Hwang et al., 2021) may be used for assisting students in achieving their learning goals with peers' power. The use of this approach can stimulate students to have more positive learning behaviours and enhanced learning motivation, which can lead them to achieve better connection between their cognitive strategies and their metacognitive and critical thinking strategy use.

Limitations of the current work reveal the need for future research. First, the current research constitutes the first step in analysing students' cognitive, metacognitive and critical thinking strategy use in an authentic, long-term learning situation in a systematic way. Certainly, more experimental studies, which may include a control group or a pre/post design to provide comparisons, are needed to confirm, better understand and extend these findings further. Second, in Pintrich's (2004) work on the MSLQ, the scales of rehearsal, elaboration, and organization reflect the use of basic cognitive and learning strategies to carry out a programming assignment and thus constitute basic indicators of cognitive regulation by students. Thus, further work is needed to study more complex rehearsal, elaboration, and organizational strategies as well as other indicators (such as forethought or reaction activities that students might use when coping with programming assignments) in more depth. Third, another important area of regulation that Pintrich distinguished has been behaviour. So the next step of our work is to explore students' self-regulation of behaviour (effort management, time management and help-seeking). Fourth, another important issue that we did not deal with in this work is context and its regulation. Regulating context implies attempts to control and regulate the tasks and learning environment context in order to enhance students' task and goal accomplishment (Corno, 1986; Kuhl, 1984; Pintrich, 2004). Our online environment follows a student-centered approach where students are responsible of uploading and evaluating their programming assignments in DSLab, so they actually take the control and regulation of their tasks, whereas they also have to decide whether they will perform the tasks individually or they will form a group and elaborate the tasks collaboratively in pairs. Consequently, students should be able to manage the climate and structure of their environment. As Zimmerman (2008) argues, when students need to achieve an efficient working environment, especially when they have to collaborate and interact with their peers at a stable base, they use specific monitoring methods that support collaborative learning through self-regulation, co-regulation, or socially shared regulation (Hadwin et al., 2011). Fifth, another important issue is regulation of motivation and affect, which includes attempts to control self-efficacy, affect and emotions through the use of various strategies (Panadero, 2017; Pintrich, 2004). Finally, an important area of research is the study of strategies to engage students in interacting and learning with DSLab platform more effectively. Indeed, high procrastinators are less successful online learners than low procrastinators (Michinov et al., 2011). Our ongoing work is exploring this issue further (Marquès, Calvet, Arguedas, Daradoumis & Mor, 2021).

ACKNOWLEDGMENTS

No acknowledgments exist.

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

PEER REVIEW

The peer review history for this article is available at https://publons.com/publon/10.1111/jcal.12605.

DATA AVAILABILITY STATEMENT

A request to access data can be directed to authors. All data have been stored in a secure place at the university database.

ETHICS STATEMENT

A request to access data can be directed to authors. All data have been stored in a secure place at the university database.

The research performed in this work is the sole work of the named authors. All student data have been de-identified after collection, so anonymized data have been used during the analysis process.

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

This article does not contain any studies with human participants or animals performed by any of the authors. Human participants (university students) have merely answered an accorded questionnaire anonymously. That is, no participants' personal data were used or recorded.

ORCID

Thanasis Daradoumis https://orcid.org/0000-0001-9362-4469
Marta Arguedas https://orcid.org/0000-0002-6609-122X

REFERENCES

Alexander, P. A., Sperl, C. T., Buehl, M. M., Fives, H., & Chiu, S. (2004). Modeling domain learning: Profiles from the field of special education. *Journal of Educational Psychology*, *96*, 545–557.

Alharbi, A., Henskens, F., & Hannaford, M. (2012). Student-centered learning objects to support the self-regulated learning of computer science. *Creative Education*, *3*, 773–783.

Alharbi, A., Henskens, F., & Hannaford, M. (2014). Personalised learning object system based on self-regulated learning theories. *International Journal of Engineering Pedagogy*, *4*(3), 24–35.

Alhazbi, S., & Hassan, M. (2013). Fostering self-regulated learning in introductory computer programming course. In *Proceedings of the 18th annual Western Canada conference on computing education* (pp. 1–4). ACM.

Andrade, M., & Evans, N. (2013). *Principles and practices for response in second language writing: Developing self-regulated learners*. Routledge.

Bannert, M., & Reimann, P. (2012). Supporting self-regulated hypermedia learning through prompts. *Instructional Science*, *40*(1), 193–211.

Barker-Plummer, D., Dale, R., & Cox, R. (2012). Using edit distance to analyse errors in a natural language to logic translation corpus. In K. Yacef, O. Zaiane, A. Hershkovitz, M. Yudelson, & J. Stamper (Eds.), *Proceedings of the Fifth International Conference on Educational Data Mining* (pp. 134 - 141). International Educational Data Mining Society.

Bergin, S., Reilly, R., & Traynor, D. (2005). Examining the role of self-regulated learning on introductory programming performance. In *Proceedings of the first international workshop on computing education research* (pp. 81–86). ACM.

Biesinger, K., & Crippen, K. (2010). The effects of feedback protocol on self-regulated learning in a web-based worked example learning environment. *Computers & Education*, *55*(4), 1470–1482.

Çakici, D. (2018). Metacognitive awareness and critical thinking abilities of pre-service EFL teachers. *Journal of Education and Learning*, *7*(5), 116–129.

Chen, C. (2002). Self-regulated learning strategies and achievement in an introduction to information systems course. *Information Technology, Learning, and Performance Journal*, *20*(1), 11–25.

Chen, H.-M., Chen, W.-H., Hsueh, N.-L., Lee, C.-C., & Li, C.-H. (2017). ProgEdu - an automatic assessment platform for programming courses. In P. Meen & Lam (Eds.), *Proceeding of the 2017 IEEE international conference on applied system innovation* (pp. 173–176). Sapporo.

Corno, L. (1986). The metacognitive control components of self-regulated learning. *Contemporary Educational Psychology*, *11*, 333–346.

Dinsmore, D. L., & Zoellner, B. P. (2018). The relation between cognitive and metacognitive strategic processing during a science simulation. *British Journal of Educational Psychology*, *88*, 95–117.

El Saadawi, G. M., Azevedo, R., Castine, M., Payne, V., Medvedeva, O., Tseytlin, E., Legowski, E., Jukic, D., & Crowley, R. S. (2010). Factors affecting feeling-of knowing in a medical intelligent tutoring system: The role of immediate feedback as a metacognitive scaffold. *Advances in Health Sciences Education*, *15*(1), 9–30.

Ergen, B., & Kanadli, S. (2017). The effect of self-regulated learning strategies on academic achievement: A Meta-analysis study. *Eurasian Journal of Educational Research*, *69*, 55–74.

Gable, R. K. (1998). Review of motivated strategies for learning questionnaire. In J. Impara & B. S. Plake (Eds.), *The thirteenth mental measurements yearbook* (pp. 681–682). The University of Nebraska Press.

Garcia, R., Falkner, K., & Vivian, R. (2018). Systematic literature review: Self-regulated learning strategies using e-learning tools for computer science. *Computers & Education*, *123*, 150–163.

Garcia, T., & Pintrich, P. R. (1992). Critical thinking and its relationship to motivation, learning strategies, and classroom experience. In *The 100th annual meeting of the American Psychological Association*. Office of Educational Research and Improvement.

Gerdes, A., Heeren, B., Jeuring, J., & van Binsbergen, L. T. (2017). Ask-Elle: An adaptable programming tutor for Haskell giving automated feedback. *International Journal of Artificial Intelligence in Education*, *27*(1), 65–100.

Grivokostopoulou, F., Perikos, I., & Hatzilygeroudis, I. (2017). An educational system for learning search algorithms and automatically assessing student performance. *International Journal of Artificial Intelligence in Education*, *27*(1), 207–240.

Hadwin, A. F., Järvelä, S., & Miller, M. (2011). Self-regulated, co-regulated, and socially shared regulation of learning. In B. J. Zimmerman & D. H. Schunk (Eds.), *Handbook of self-regulation of learning and performance* (pp. 65–84). Routledge.

Halpern, D. F. (1998). Teaching critical thinking across domains: Dispositions, skills, structure training, andmetacognitive monitoring. *American Psychologist*, *53*(4), 449–455.

Hundt, C., Schlarb, M., & Schmidt, B. (2017). SAUCE: A web application for interactive teaching and learning of parallel programming. *Journal of Parallel and Distributed Computing*, *105*(2017), 163–173.

Hwang, G. J., Wang, S. Y., & Lai, C. L. (2021). Effects of a social regulation-based online learning framework on students' learning achievements and behaviors in mathematics. *Computers & Education*, *160*, 104031.

Ifenthaler, D. (2012). Determining the effectiveness of prompts for self-regulated learning in problem-solving scenarios. *Educational Technology & Society*, *15*(1), 38–52.

Ihantola, P., Ahoniemi, T., Karavirta, V., & Seppälä, O. (2010). Review of recent systems for automatic assessment of programming assignments. In *Proceedings of the 10th Koli calling international conference on computing education research* (pp. 86–93). ACM.

Isaacson, R., & Fujita, F. (2006). Metacognitive knowledge monitoring and self-regulated learning: Academic success and reflections on learning. *Journal of the Scholarship of Teaching and Learning*, *6*(1), 39–55.

Kasimi, Y. (2012). Cognitive and metacognitive strategies employed by Iranian and Turkish EFL readers. *International Association of Research in Foreign Language Education and Applied Linguistics*, *1*(3), 159–174.

Kauffman, D. F., Zhao, R., & Yang, Y.-S. (2011). Effects of online note taking formats and self-monitoring prompts on learning from online text: Using technology to enhance self-regulated learning. *Contemporary Educational Psychology*, *36*(4), 313–322.

Kelleher, J. (2014). Employing Git in the classroom. In *Computer applications and information systems (WCCAIS), 2014 world congress* (pp. 1–4). IEEE.

Kramarski, B., & Gutman, M. (2006). How can self-regulated learning be supported in mathematical E-learning environments? *Journal of Computer Assisted Learning*, *22*(1), 24–33.

Ku, K. Y. L., & Ho, I. T. (2014). Metacognitive strategies that enhance critical thinking. *Metacognition and Learning*, *5*(3), 251–267.

Kuhl, J. (1984). Volitional aspects of achievement motivation and learned helplessness: Toward a comprehensive theory of action control. In B. Maher & W. Maher (Eds.), *Progress in experimental personality research (vol. 13)* (pp. 99–171). Academic Press.

Kuhn, D., & Dean, D. (2004). Metacognition: A bridge between cognitive psychology and educational practice. *Theory Into Practice*, *43*(4), 268–274.

Lai, C. L., Hwang, G. J., & Tu, Y. H. (2018). The effects of computer-supported self-regulation in science inquiry on learning outcomes, learning processes, and self-efficacy. *Educational Technology Research and Development*, *66*(4), 863–892.

Magno, C. (2010). The role of metacognitive skills in developing critical thinking. *Metacognition and Learning*, *5*(2), 137–156.

Marqués, J. M., Daradoumis, T., Calvet, L., & Arguedas, M. (2020). Fruitful Student Interactions and Perceived Learning Improvement in DSLab: a Dynamic Assessment Tool for Distributed Programming. *British Journal of Educational Technology*, *51*(1), 53–70.

Marqués, J. M., Calvet, L., Arguedas, M., Daradoumis, T. & Mor, E. (2021). Using a Notification, Recommendation and Monitoring System to Improve Interaction in an Automated Assessment Tool: An Analysis of Students' Perceptions, *International Journal of Human-Computer Interaction*. https://doi.org/10.1080/10447318.2021.1938400.

Manso, M., Caeiro, M., & Llamas, M. (2016). Are learning software systems well-prepared to support self-regulated learning strategies? *International Journal of Engineering Education*, *32*(2), 1015–1023.

Michinov, N., Brunot, S., Le Bohec, O., Juhel, J., & Delaval, M. (2011). Procrastination, participation, and performance in online learning environments. *Computers & Education*, *56*(1), 243–252.

Naimnule, L., & Corebima, A. D. (2018). The correlation between metacognitive skills and critical thinking skills toward students' process skills in biology learning. *Journal of Pedagogical Research*, *2*(2), 122–134.

Panadero, E. (2017). A review of self-regulated learning: Six models and four directions for research. *Frontiers in Psychology*, *8*(422), 1–28.

Panadero, E., & Järvelä, S. (2015). Socially shared regulation of learning: A review. *European Psychologist*, *20*, 190–203.

Pedrosa, D., Cravino, J., Morgado, L., & Barreira, C. (2016). Self-regulated learning in computer programming: Strategies students adopted during an assignment. In C. Allison, L. Morgado, J. Pirker, D. Beck, J. Richter, & C. Gütl (Eds.), *Immersive learning research network. Communications in Computer and Information Science (Vol. 621)* (pp. 87–101). Springer.

Pedrosa, D., Cravino, J., Morgado, L., & Barreira, C. (2019). Co-regulated learning in computer programming: Students co-reflection about learning strategies adopted during an assignment. In M. Tsitouridou, J. A. Diniz, & T. A. Mikropoulos (Eds.), *Technology and innovation in learning, teaching and education (TECH-EDU 2018)* (pp. 13–28). Springer Nature Switzerland AG.

Pettit, R. S., Homer, J. D., Holcomb, K. M., Simone, N., & Mengel, S. A. (2015). Are automated assessment tools helpful in programming courses? In *Proceedings of the 122nd ASEE Annual Conference & Exposition* (pp. 1–20). Seattle, WA.

Pintrich, P. R. (2004). A conceptual framework for assessing motivation and selfregulated learning in college students. *Educational Psychology Review*, *16*, 385–407.

Pintrich, P. R., & De Groot, E. V. (1990). Motivational and self-regulated learning components of classroom academic performance. *Journal of Educational Psycology*, *82*(1), 33–40.

Rivers, K., & Koedinger, K. R. (2017). Data-driven hint generation in vast solution spaces: A self-improving Python programming tutor. *International Journal of Artificial Intelligence in Education*, *27*(1), 37–64.

Robinson, P. E., & Carroll, J. (2017). An online learning platform for teaching, learning, and assessment of programming. In *Global engineering education conference* (pp. 547–556). IEEE.

Saeedzadeh, M., Raeisoon, M. R., & Mohammadi, Y. (2018). The relationship between cognitive and metacognitive strategies and academic achievement of students of Birjand University of Medical Sciences. *Future of Medical Education Journal*, *8*(1), 27–30.

Schneider, W., & Pressley, M. (1997). *Memory development between 2 and 20*. Erlbaum.
Stajduhar, I., & Mausa, G. (2015). Using string similarity metrics for automated grading of SQL statements. In *Proceedings of the 38th IEEE international convention on information and communication technology, electronics and microelectronics (pp. 1250–1255)*. IEEE.
Sukhoroslov, O. (2018). Building web-based Services for Practical Exercises in parallel and distributed computing. *Journal of Parallel and Distributed Computing*, *118*(1), 177–188.
Suleman, H. (2008). Automatic marking with Sakai. In *Proceedings of the 2008 annual conference of the south African Institute of Computer Scientists and Information Technologists on IT research in developing countries* (pp. 229–236). ACM.
Taylor, R. T. (2012). Review of the motivated strategies for learning questionnaire (MSLQ). Using reliability generalization techniques to assess scale reliability (doctoral dissertation). Auburn University.
van Ewijk, C. D. (2011). Assessing students' acquisition of self-regulated learning skills using meta-analysis. In B. J. Zimmerman & D. H. Schunk (Eds.), *Handbook of self-regulation of learning and performance* (pp. 376–390). Taylor & Francis.
Vujošević-Janičić, M., Nikolić, M., Tošić, D., & Kuncak, V. (2013). Software verification and graph similarity for automated evaluation of students' assignments. *Information and Software Technology*, *55*(6), 1004–1016.
Wang, C. H., Shannon, D., & Ross, M. (2013). Students' characteristics, self-regulated learning, technology self-efficacy, and course outcomes in online learning. *Distance Education*, *34*(3), 302–323.
Wäschle, K., Lachner, A., Stucke, B., Rey, S., Frömmel, C., & Nückles, M. (2014). Effects of visual feedback on medical students' procrastination within web-based planning and reflection protocols. *Computers in Human Behavior*, *41*, 120–136.
Weinstein, C., & Mayer, R. (1986). The teaching of learning strategies. In M. Wittrock (Ed.), *Handbook of research on teaching and learning* (pp. 315–327). Macmillan.
Weinstein, C. E., Husman, J., & Dierking, D. R. (2000). Self-regulation interventions with a focus on learning strategies. In M. Boekaerts, P. Pintrich, & M. Seidner (Eds.), *Self-regulation: Theory, research, and applications* (pp. 727–747). Academic Press.
Wong, J., Baars, M., Davis, D., van der Zee, T., Houben, G. J., & Paas, F. (2019). Supporting self-regulated learning in online learning environments and MOOCs: A systematic review. *International Journal of Human-Computer Interaction*, *35*(4–5), 356–373.
Yeh, Y. F., Chen, M. C., Hung, P. H., & Hwang, G. J. (2010). Optimal self-explanation prompt design in dynamic multi-representational learning environments. *Computers & Education*, *54*(4), 1089–1100.
Zheng, L. (2016). The effectiveness of self-regulated learning scaffolds on academic performance in computer-based learning environments: A meta-analysis. *Asia Pacific Education Review*, *17*(2), 187–202.
Zheng, X. L., Kim, H. S., Lai, W. H., & Hwang, G. J. (2020). Cognitive regulations in ICT-supported flipped classroom interactions: An activity theory perspective. *British Journal of Educational Technology*, *51*(1), 103–130.
Zimmerman, B. J. (2008). Investigation self-regulation and motivation: Historical background, methodological developments, and future prospects. *American Educational Research Journal*, *45*(1), 166–183.
Zimmerman, B. J. (2013). From cognitive modeling to self-regulation: A social cognitive career path. *Educational Psychologist*, *48*(3), 135–147.
Zimmerman, B. J., & Martinez-Pons, M. (1986). Development of a structured interview for assessing student use of self-regulated learning strategies. *American Educational Research Journal*, *23*, 614–628.

How to cite this article: Marquès Puig, J. M., Daradoumis, T., Arguedas, M., & Calvet Liñan, L. (2022). Using a distributed systems laboratory to facilitate students' cognitive, metacognitive and critical thinking strategy use. *Journal of Computer Assisted Learning*, *38*(1), 209–222. https://doi.org/10.1111/jcal.12605