Exploring the Association Between Self-Regulation of Learning and Programming Learning: A Systematic Literature Review and Multinational Investigation

Leonardo Silva
University of Coimbra
Coimbra, Portugal
leonardo.silva@garanhuns.ifpe.edu.br

António Mendes
University of Coimbra
Coimbra, Portugal
toze@dei.uc.pt

Anabela Gomes
Coimbra Polytechnic - ISEC
Coimbra, Portugal
anabela@isec.pt

Gabriel Fortes
Facultad de Psicología
Universidad Alberto Hurtado
Santiago, Chile
gfortes@uahurtado.cl

Cahn Tong Lam
Macao Polytechnic Institute
Macao, China
ctlam@ipm.edu.mo

Calana Chan
Macao Polytechnic Institute
Macao, China
calanachan@ipm.edu.mo

***Abstract*—This Research Full Paper presents a collection of evidences about the association between self-regulation variables and programming learning. Researchers have been investigating this thematic and despite the apparent benefits, it is necessary to summarize the published evidence and provide a new collection of them, which this study seeks to contribute. An observational investigation was performed in two countries with fifty-nine students, who had their SRL and programming learning metrics collected and correlated. Moreover, a systematic literature review was also conducted, and the existing evidence summarized. The results support an association between metacognitive and motivational regulatory strategies with programming learning but do not support for cognitive strategies. Our analysis shows a need for more studies to provide a solid body of knowledge on this thematic.**

***Index Terms*—self-regulated learning, programming learning, programming education**

I. Introduction

Programming students often struggle during the learning process [1]. Several factors contribute to this difficulty, including the lack or inefficient use of learning strategies [2]. Students who orchestrate the use of cognitive strategies, recognise their difficulties and seek ways to overcome them, controlling their emotions and motivations are more likely to succeed academically [3]. These students are defined as self-regulated learners due to their ability to employ efforts and monitor their actions (properly regulating them) to achieve educational objectives.

An association between Self-Regulation of Learning (SRL) and improvements in academic performance has been found in multiple educational contexts [3]. SRL has consolidated itself as a fundamental educational theory to describe the factors contributing to the educational process [4].

Programming educational researchers were interested in studying how SRL constructs applies to this academic field [5]. This educational domain has unique characteristics that demand intense use of cognitive and metacognitive resources, which students' self-regulation can favour, requiring more investigation to understand the nuances of SRL in programming [6].

One research area investigates how SRL associates with programming learning, and the initial evidence points to a relationship between students' academic performance and their SRL capabilities [7]. Still, more investigations are needed to consolidate this knowledge [6]. Another research area investigates SRL strategies specific to the programming education domain. However, further investigations are necessary to understand the specificities of this learning domain under the theoretical foundations of SRL [6], [8]. This paper aims to contribute to both of these areas.

This work investigates the association between students' self-regulation of learning scores and their programming learning performance. The following research questions were developed to achieve this objective: a) What is the current state of the art regarding the association between SRL and programming learning? b) What is the association between the general self-regulation scores and the programming achievement scores of the students who participated in the investigation? c) What is the association between students' programming-specific SRL scores and programming achievement scores of the students who participated in the investigation? d) What are the explanatory variables of programming-specific SRL and general-purpose strategies that influence the academic performance of the students who participated in the investigation?

This study employed two methodological approaches to answer the aforementioned research questions. First, a systematic literature review was performed, followed by exploratory quantitative research. The latter included fifty-nine students from two different educational institutions (Brazil and Macao), in which their academic data was analysed. The objective is to compile the findings in the literature and provide new evidence concerning this phenomenon.

The main contributions of this study are three-fold. First, it summarises the literature regarding the association between SRL and programming performance found in the current state of the art. Secondly, we provide supporting evidence on this topic using multinational data. Finally, our methodology replicates previous studies, which is a necessary practice of the scientific method, also providing a novelty approach by applying an instrument that measures programming-specific regulation of learning strategies, little explored in the literature.

This study is organised as follows: Section II reviews the literature related to SRL and programming education. Section III details the methodological procedures adopted in this investigation. Section IV presents the systematic literature review and observational study results. Finally, Section V discuss the results and their associations with current research.

II. Literature review

It is known that programming learning is a complex process that frequently results in dropout or unmotivated students [9]. Due to its association with abstract concepts, the use of multiple cognitive strategies, and the need to constantly stay motivated to face difficulties [2], self-regulation of learning (SRL) has the necessary characteristics to support this process [6].

SRL is an educational theory that has become central to describing learning practices [3]. Studies on this topic understand that the educational process involves realising and controlling cognitive, metacognitive, motivational, emotional, and behavioural actions [10]. Students' who regulate their actions on these areas are more likely to succeed [11]. For example, it is known that students who plan their learning routine, establish goals, monitor their performance identifying situations for improvements, knowing how to deal with failure situations, among others, increase their chance of academic success [12].

There is increasing evidence concerning the importance of SRL skills to academic success [13], in particular in complex educational domains such as programming education [6]. It has motivated programming education researchers to ground their investigations' on the self-regulated learning theory [5].

Authors found that high-performing programming students present more complex self-regulated behaviour than low-performing students, frequently using regulatory actions, such as planning and monitoring during algorithm development [14], [15].

Despite the apparent benefits, there is a need to provide supporting evidence regarding the association of SRL with programming learning. It could be provided mainly by two methodological approaches, experimental and observational studies. The former was summarised in a previous systematic literature review performed by our research group [5]. Several interventions grounded on SRL constructs were analysed, and the majority of interventions reviewed by the authors proved to be effective not only in stimulating SRL, but also programming learning.

Observational studies are also used to identify SRL and programming learning associations [16]. These studies measure programming students' SRL skills, usually using standardised questionnaires and calculating their correlation with programming metrics, such as performance, engagement, and others [6]. This type of study is also interested in analysing how the multiple dimensions presented in the questionnaires explain the variance in the performance. For example, Bergin et al. found that metacognitive and managerial strategies were frequently present in high performing students, whereas cognitive strategies were not related to programming performance [7]. These studies are of interest in our investigation, and their analysis will be detailed in Section IV.

Studies usually use general-purpose instruments to measure SRL strategies. These instruments are widely used in multiple academic domains and seek to measure learning strategies regardless of the educational context [17]. Despite its relevance, SRL is known to have specificities, depending on the educational context [3], [13]. Consequently, it is also of interest to investigate how specific programming regulation of learning strategies are associated with academic performance, which is currently lacking in the existing literature, and this study also seeks to investigate.

III. Methodology

This work adopted a methodological approach consisting of two phases. In the first, a systematic literature review and meta-analysis were performed to summarise state of the art regarding the association between SRL and programming learning. This knowledge allowed us to identify the current gaps and the methodological procedures adopted, grounding our proposition to replicate previous investigations and also corroborating the relevance of investigating the association of specific programming regulatory strategies with performance, which was not found in the literature. In this respect, an observational study was carried out to provide our evidence on this thematic.

A. Systematic Literature Review

This study followed Kitchenhams' methodology to perform the literature review [18], due to its adoption in computer science research. The following research questions guided this process: a) What is the current state of the art regarding the association between SRL and programming learning?; b) What were the methodological procedures adopted by the studies in this research area?.

The inclusion criteria used were: i) only observational studies in the context of introductory programming, ii) grounded on SRL theory, and iii) measured SRL and metrics associated with programming learning. Not only metrics directly related to learning outcomes, such as course grades, were considered, but also other variables, such as motivation and engagement. The digital libraries of ACM, IEEE, Science Direct and Eric were searched with the keywords: [self-regulat* OR SRL OR metacognit* OR co-regulat* OR SSRL OR" regulation of learning"] AND ["learning programming" OR" programming learning" OR" teaching programming" OR" programming teaching" OR" novice programmer" OR" programming education"]. These libraries and the keywords were used due to their relevance to the research topic. We did not filter papers by date and considered published results until March 2021.

The search strategy consisted of two stages. First, keywords were used to search in the libraries mentioned, and the title, abstract, and keywords of identified studies were manually examined to assess their adequacy to the inclusion criteria. The papers admitted were selected for full reading (second stage), and the inclusion criteria were again analysed. Finally, selected papers had the following data extracted: title, study objectives, statistics, sample size and characteristics, and educational context.

Meta-analysis with a random-effects model was performed for the scales with at least two studies that reported it. This statistical procedure combines the results from multiple studies providing a more reliable estimate of the size of an effect. The metacor function from R was used.

B. Observational Study

A total of fifty-nine programming students participated in the study. Twelve enrolled in an associate degree course in computing with a three-year duration from a Brazilian university, and forty-seven from Macao enrolled in a four-year duration BSc in Computing. Both programming courses had a similar syllabus, contemplating variables, conditionals, loops, methods and arrays. Brazilian classes were taught in Python and Java, whereas Macao students' classes used Java, totalling 80 and 45 hours in the semester, respectively.

1) Instruments: Students' answered the Motivated for Learning Strategies Questionnaire (MSLQ), developed by Pintrich et al. [17], with a seven-point Likert scale. This instrument was chosen because it is one of the most reliable and used to measure SRL skills in multiple educational domains [4], including programming [6]. The instrument has two versions, one with 81 and another with 44 items, measuring students' motivation and self-regulated strategies. This questionnaire is divided into three constructs, motivation, cognitive and metacognitive strategies. We opted to use the short version as it fits our needs, and only the scales of cognitive and metacognitive strategies were used because one of the objectives of this study is to compare the results to the SPSQ questionnaire (described below), which focuses only on these constructs. The cognitive strategies mentioned in the instrument relates to students' capability to organise their learning materials, ability to focus and comprehend the relevant parts of a text, perform annotations to organise learning ideas, among others. The metacognitive strategies focus on students' ability to monitor and plan their learning and the capability to overcome their difficulties. According to the MSLQ authors, it is possible to use parts of the instrument [17].

Students also answered the Self-Regulated Programming Strategies Questionnaire (SPSQ) developed by Lu et al. [19], with a five-point Likert scale. The use of a context-specific SRL instrument is relevant to evaluate programming specificities [6]. This instrument is composed of 23 items that measure students' 1) attention regulation - ATTR (e.g. *When implementing a program, I focused on programming so that I would not be distracted by other things*); 2 planning ahead of programming - PA (e.g. *Before implementing a program, I designed a flowchart of what I needed to address in the program*); 3) content monitoring - CM (e.g., *When implementing a program, I checked to see whether what I had implemented was correct*); 4) organisation; 5) checking and correcting - CC (e.g., *At the end of implementation, I went back and read the code of the module to make sure it was OK*); 6) planning during coding - PDC (e.g., *When implementing the program, I thought about what I was going to implement next*); and 7) self-evaluation - SE (e.g. *After I implemented a program module, or part of it, I thought about whether what I had implemented was correct*). As the organisation scale focus on requirements design and documentation, which are out of the programming courses' scope, their items were not used. The SPSQ instrument mainly focuses on cognitive and metacognitive programming strategies. For example, how the student prepares to solve a programming problem, monitors whether their algorithm is adequate, and the students' reflective actions on improving his development practices.

The final grade in the introductory programming curricular component was used (0 to 100) in the correlation calculus and regression analysis. When data followed a normal distribution and had lower outliers, the Pearson coefficient was used. On the contrary, the Spearman correlation was used because it is less sensitive to these assumptions [20]. Regression analysis was performed employing stepwise regressions. A p-value less than 0.05 was considered to state the association as statistically significant.

IV. Results

The results will be presented following the research questions order.

A. Research Question a)

In the systematic literature review process, ninety-two papers were selected for a complete reading. After analysing the inclusion criteria, six studies were included in the review (Table I). The main causes for exclusion were: not in the context of introductory programming, did not calculate the association of SRL and programming metrics, and not theoretically grounded on SRL constructs.

Studies mainly focused on associating SRL variables with programming performance and engagement (Table II). Tian and Wu measured learning engagement, being the only study that did not use programming achievement [21]. The authors used the Schaufeli et al. instrument that conceptualises

TABLE I

STUDIES INCLUDED IN THE LITERATURE REVIEW.

Study	**Description**	**Instrument**	**Educational level**
[21]	Evaluates the association of programming students SRL with learning engagement.	Motivational regulation questionnaire	Higher education
[7]	Sought to identify how regulation strategies are associated with programming learning performance.	MSLQ	Higher education
[22]	Used SRL variables as predictors for programming performance.	Learner Self-Regulation Scale	Vocational college
[23]	Developed a model of SRL variables used to predict programming performance.	Own instrument adapted from others	Vocational college
[24]	Sought to identify multiple factors affecting introductory programming performance, including SRL	MSLQ	Higher education
[25]	Focused on exploring the association of motivational aspects (under the framework of SRL) with programming performance	MSLQ	Higher education

TABLE II

CORRELATIONS BETWEEN SRL CONSTRUCTS AND PROGRAMMING OUTCOMES. NS = NON-SIGNIFICANT; SE = SELF-EFFICACY; IM = INTRINSIC MOTIVATION; GC = GOAL COMMITMENT; SS = SELF-SATISFACTION; TA = TEST-ANXIETY; MS = METACOGNITIVE STRATEGIES; CS = COGNITIVE STRATEGIES; RS = RESOURCE STRATEGY; IT = INTERACTIVE IN THE ONLINE LEARNING ENVIRONMENT; PS = PERCEIVED SATISFACTION; PU = PERCEIVED USEFULNESS; LSR = LEARNER SELF-REGULATION; RV = REGULATION OF VALUE; RP = REGULATION OF PERFORMANCE GOALS; SC = SELF-CONSEQUATING; ES = ENVIRONMENTAL STRUCTURING; RSI = REGULATION OF SITUATIONAL INTEREST; RM = REGULATION OF MASTERY GOALS.

Study	**Outcome**	**SE**	**IM**	**GC**	**SS**	**TA**	**MS**	**CS**	**RS**	**IT**	**PS**	**PU**	**LSR**	**RV**	**RP**	**SC**	**ES**	**RSI**	**RM**
[21]	Vigor													.54	.65	NS	.41	.51	.45
[21]	Dedication													.64	.71	NS	.45	.62	.57
[21]	Absorption													.58	.65	NS	.44	.62	.56
[7]	Grade						.54	NS	.55	NS	NS	.12	NS						
[22]	Grade	.18				NS													
[23]	Grade	.39		.32	.32														
[24]	Grade	.30	NS			- .38													
[25]	Grade																		
This work	Grade						.26	NS											
Effect-size		.31					.39												

engagement into the dimensions of vigour, dedication and absorption. In the context of Tian and Wu investigation, vigour relates to students' willingness to invest effort in their learning, dedication refers to the student's sense of learning, and absorption is the degree to which students' concentrate on their work to overcome their difficulties and achieve the learning objectives. The rest of the papers investigated the association of SRL skills with programming learning using course grades as a dependent variable.

Two lines of research grounded on SRL were observed on the rest of papers, one interested in motivational aspects, in special self-efficacy and the other investigated the influence of cognitive and metacognitive strategies. Consequently, some authors have not used the instruments fully and opted for the scales associated with their research objectives. For example, [24] used only the MSLQ scales associated with motivation, whereas our study and [7] focused on the scales related to cognitive and metacognitive strategies. A variety of instruments were applied to measure SRL, with MSLQ being the most used by 42.8% of studies.

Several statistical procedures were applied to identify the association of SRL with programming metrics, such as correlations and regression analysis. Regression analysis and Structural Equation Modeling (SEM) were employed to identify the variables that mainly explained the students' grades variance and predict students' performance. The findings support the adequacy of the MSLQ scales as predictive factors. Bergin et al. applied a stepwise regression finding that metacognitive strategies and resource management accounted for 29% of the variance [7]. On the model developed by [22], the authors found that only self-efficacy contributed significantly to the grades. Kuo et al. [23], and Lishinski et al. [25] applied a slightly different approach using SEM to build an exploratory model, finding that self-efficacy and setting goals scores are related to programming achievement.

B. *Research Questions b) and c)*

Fifty-nine students answered the MSLQ and SPSQ instruments. Forty-seven from Macao (mean age: 20.51, gender: 85.1% male, 8.6% prefer not to say, and 6.3% female) and twelve from Brazil (mean-age: 21.25, gender: 75% male and 25% female). The Cronbach alpha for the MSLQ instrument was 0.92 [0.89, 0.95], and for the SPSQ was 0.90 [0.87, 0.93].

Students' scores on MSLQ and SPSQ, in conjunction with their correlation with programming performance, are presented in Table III. Results are presented categorised by the educational institution, gender, and the overall result, combining the Brazilian and Macanese scores. Figure 1 presents a graphic with the average students' scores on the SPSQ dimensions.

TABLE III
DESCRIPTIVE STATISTICS FOR MSLQ, SPSQ AND PROGRAMMING PERFORMANCE.

	Ed. institution - Mean (SD)		**Gender - Mean (SD)**			**Overall**
	Brazil (n=12)	Macao (n=47)	Male (n=49)	Female (n=6)	Prefer not to say (n=4)	
MSLQ	5.5 (0.39)	4.76 (0.83)	4.72 (0.72)	5.68 (0.49)	6.04 (1.14)	4.97 (0.85)
SPSQ	4.1 (0.36)	3.7 (0.52)	3.76	3.99	4.41	3.51 (0.53)
Performance	89.41 (10.49)	57.48 (23.63)	64.15 (25.28)	83 (10.63)	73.25 (22.27)	65.87 (25.36)
SPSQ correlation with performance	0.55	0.43	0.48	Non-significant	Non-significant	0.41
MSLQ correlation with performance	Non-significant	0.58	0.64	0.82	Non-significant	0.63

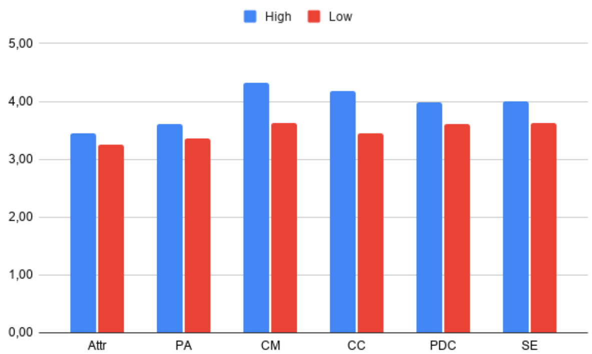

Fig. 1. SPSQ scores for high and low performing students.

Statistical analyses were performed to assess differences in scores when grouped by the educational institution and gender (three factors: male, female, and prefer not to say), allowing the observation of whether the samples differ based on these variables. Students' scores on MSLQ and SPSQ followed a normal distribution (Shapiro-Wilk test). A statistically significant result was found using t-test for MSLQ (t = 2.6854 p-value <0.01) and SPSQ (t = 3.2833 p-value <0.001) when grouped by university, with Brazilian students presenting a higher score on both instruments. Scores were also different when grouped by gender using One-way ANOVA, for MSLQ (F = 9.53 p-value <0.001) and SPSQ (F=4.40 p-value <0.01). The post-test of Tukey HSD was performed, showing a difference between the scores of male and female students (on MSLQ) and between students who did not prefer to say their gender and males (on MSLQ and SPSQ).

The association between the MSLQ and SPSQ scores with programming learning, measured by the student's final grade, was also calculated. This procedure used data organised by the educational institution, gender and also overall scores. A non-significant correlation was identified for some of them, whereas it presented a significant variation for others (Table III).

We have also calculated the correlation between the scales of MSLQ (cognitive and metacognitive strategies) and SPSQ (attention regulation, planning ahead of programming, content monitoring, checking and correcting, planning during coding and self-evaluation) with programming performance (Table IV). Some of these scales were correlated, whereas, in others, the correlation was not significant (e.g., cognitive strategies in the MSLQ).

C. Research Question d)

Stepwise multiple regression analysis was performed to identify the association between MSLQ and SPSQ with programming achievement. Two procedures were performed; first, the scores on these instruments were used as predictors and the course grade as a dependent variable. Only the MSLQ scores were statistically significant and accounted for 38.6% of the variance in students' grades (p-value <0.0001). Secondly, we have developed two regression models, one using the SPSQ scales and the other the MSLQ scales as predictors. MSLQ scales were not adequate to be used in the model (p-value >0.05 and low R-squared). For SPSQ, only the scales of content monitoring and checking and correcting from SPSQ were adequate to the model, explaining 41.6% of the variance on students' grade.

A further analysis was performed to comprehend better the association of MSLQ and SPSQ on programming learning. Students were grouped according to their performance as low (n = 20) and high-performing (n=39) (those who failed and succeeded in the course). We built a regression model for each group, considering the SPSQ and MSLQ scales as predictors of academic performance. This procedure allows us to understand whether the association between the scales and the learning score differs between the groups analysed so that it is possible to understand which are more explanatory for students with better performance, for example.

The MSLQ scales were not good predictors for high and low performing students. Using SPSQ scales, for high-performing students, it was identified that the scores on planning ahead of programming, checking and correcting, and self-evaluation were the best predictors and accounted for 32.9% of the variance on students' scores. The identified predictors for low-performing students were planning ahead of programming, attention regulation, and self-evaluation, which accounted for 34.6% of the variance.

V. DISCUSSIONS

The discussion of the results will be presented following the research questions order.

A. Research question a)

Before analysing the findings in the review, it should be noted that there was a significant variety of the instruments used, making it more challenging to provide an overview of the topic. Even when the same instrument was applied, the authors

TABLE IV
CORRELATION BETWEEN MSLQ AND SPSQ SCALES WITH PROGRAMMING PERFORMANCE.

	MSLQ	**SPSQ**			
	Metacognitive strategies	Content monitoring	Checking and correcting	Planning during coding	Self-evaluation
Programming learning	0.26	0.56	0.64	0.32	0.42

used different scales according to their research objectives, corroborating the previous statement about the complexity of providing a general analysis.

There is an extensive body of research about the relevance of self-efficacy (SE) to the learning process [26]., Three studies that (SE) with programming learning reported significant results. The models built with regression analysis showed the influence of SE on the variance of students' grades. The meta-analysis indicate an effect-size of 0.31 which is considered as medium for educational studies [27], suggesting the relevance of this construct to the programming learning domain, corroborating the previous finding on this thematic [28].

Intrinsic value, a motivational construct, did not associate with programming learning, which is an interesting finding since it is usually related to educational achievement [29], [30]. While motivation is a complex phenomenon, one possible response to this contradictory finding is related to differences in quantity and quality of motivation in self-regulated learning strategies [11], [12]. While most questionnaires emphasise motivation as a quantity of will that drives students' towards achieving a learning goal, we should keep in mind that there are differences in quality, such as dispositions to learning goals, goal-achievement, task avoidance, mastery and many other "subtypes" of learning behaviour that have already shown to highly correlate to SRL strategies [31].

The other SRL scales, such as resource strategy management, were measured by only one study and deserve further investigations to understand its association with programming learning better. We discuss the results for cognitive and metacognitive strategies in the next section, along with the results from our investigation.

In conclusion, it is observed that two general-use regulation of learning dimensions have more evidence about their association with learning to program: self-efficacy and the use of metacognitive strategies. In comparison, cognitive strategies and intrinsic value were not correlated, which contradicts traditional research on these aspects [2]. Our observational study corroborates these findings, and additional analysis is necessary to comprehend this issue. The rest of the scales, although pointing to an association, need more supporting evidence.

B. *Research question b) and c)*

A moderated correlation between overall MSLQ and SPSQ scores with programming performance was identified (0.63 and 0.41, respectively). The findings corroborate the association of SRL scores measured by MSLQ with academic achievement, as also observed in other learning domains [3]. Combining these findings with those presented in Section V-A, and in previous literature, as reported in Section II, it appears that SRL is relevant programming education. SRL contemplates a multitude of learning strategies, some of them, such as metacognition, is widely recognised as relevant to the programming learning process [1], [2], which can be an explanatory factor for that.

We have also used the scores on the MSLQ scales individually to understand its association with programming performance. Our findings support those from Bergin et al. [7], as we did not find a statistically significant correlation between cognitive strategies and programming achievement. On the other hand, the use of metacognition strategies correlated with performance. The identified effect-size of 0.39 can be interpreted as moderated for educational studies [27], highlight the relevance of metacognition for programming education.

However, the aforementioned result posits a challenge to explain whether this finding is specific to our and [7] data sets or is a programming education specificity. We might hypothesise that programming, different to other learning contexts, depends much more on metacognitive monitoring (such as coding review process) than concept correctness or memory recall (related to cognitive strategies). However, these assumptions require further empirical investigation. The discussion over domain-specific and domain-general [11], [12] might be essential to understand if programming education has specific self-regulated strategies and if these transfer from context to context. If so, the general-purpose learning regulation skills generally measured in SRL questionnaires might not be fully appropriated.

Although the correlation between the SPSQ scores and programming learning was lower than the MSLQ correlation, it points to an instrument with potential for use when the objective is to measure specific constructs of regulation in programming. However, we emphasize the need for more evidence on the use of this questionnaire.

The analysis using the SPSQ scales indicates that only content monitoring, checking and correcting, planning during coding, and self-evaluation are associated with programming learning. These items are associated with metacognitive actions, and this finding is consistent with the literature [2]. Previous findings demonstrate that students who perform poorly on monitoring and reflecting on their learning are more likely to fail in the programming educational process [32]–[36].

Planning ahead of coding was not statistically significant with learning performance, presenting a low correlation. This result was surprising as students difficulties with algorithm planning have been reported to negatively impact the algorithm development process [14], [37]. The items on these categories are classified as cognitive strategies, and this result supports the finding from MSLQ that shown no association of these strategies with programming performance. As previously reported, it is necessary to understand which cognitive strategies are effective for learning programming and if the items used in the questionnaires are adequate to represent them. We believe the items related to cognitive strategies and used in SPSQ might not match the strategies executed by students. For example, the student is asked if he/she draws a flow chart to plan their algorithm (it is assumed that this is a type of planning). However, the student may not do this but do other actions associated with planning.

When the MSLQ and SPSQ scores were grouped by educational instituions, differences were statistically significant. These variations are expected and could be explained by several factors, such as culture, the pedagogical structure of the educational institutions, socioeconomic status, among other factors [3]. As reported by Ambrósio and Martins, who investigated the development of SRL over programming students from two different countries, pedagogical differences among institutions might account for the variability in the results [38]. We could not provide further analysis on this subject with our data, which deserves a future investigation.

It was also observed how the scores differ among students of the same class, corroborating the understanding that the programming classes are heterogeneous [2]. This implies, in many cases, that the teacher needs to deal with situations where students have a high power to regulate themselves, and in other cases, they need support to accomplish this. In this sense, it is assessed that the use of technologies can be a way to mitigate this problem by providing adequate regulatory support to students to develop these skills.

The scores on the instruments were also different when students were grouped by their gender. Male students had the lowest score on MSLQ and SPSQ, differing significantly from females and students who did not inform their gender. Also, the association between MSLQ and programming learning was higher for females than males. Gender differences in previous literature are inconsistent, [21] reported a higher score from girls on MSLQ, but [25] did not find differences. Virtanen and Nevgi observed in higher education students that gender differences were also related to disciplinary content [39]. Nonetheless, further empirical investigation is needed to understand what facilitates SRL in female students in technology in general and in programming specifically.

C. Research question d)

As demonstrated in Section IV-A, multiple MSLQ scales proved to be adequate predictors for performance in introducing programming. Our result differs, as the scales of metacognition and cognition strategies were not adequate.

Analysing SPSQ scales as predictors, our stepwise regression found that only content monitoring and checking and correcting were adequate. These scales are associated with monitoring and evaluative actions students' perform during algorithm development and are reported in the literature as crucial regulatory skills [14]. Our result corroborates the association of these actions with better academic performance.

Lastly, we have also performed a slightly different approach, using the MSLQ and SPSQ overall score to build a regression model, but only MSLQ was adequate. Future investigation should replicate the use of the SPSQ instrument to assess its adequacy as a predictive factor of students' performance.

D. Implication for research and practice

This study summarises the evidence regarding SRL association with programming learning. The majority of the previous findings were corroborated, strengthening the argument about the relevance of SRL for programming education. Researchers and educators could explore this phenomenon in programming education by considering the relevance of self-efficacy and metacognitive strategies but also exploring how the other scales are associated (or not) with programming learning. This work also highlights a new perspective on investigating SRL using programming-specific instruments, which is crucial to understand how the specificities of programming education could be studied under the umbrella of SRL. However, some weaknesses are still present in state of the art and which compromise a more assertive assessment of the influence of SRL on programming learning. Two problems were identified in this review: firstly, the small number of studies identified and the fragmentation in how SRL was measured, which, when combined with the previous problem, resulted in cases in which there is only a single study that investigated a specific dimension of SRL. Other difficulties were detailed in our previous study, in particular the issues associated with experimental design.

E. Conclusion

Self-regulation of learning is a construct considered necessary for the learning process. We investigated how this phenomenon is associated with programming learning by summarising the findings in the literature and performing an observational study.

From the results obtained from the programming classes in Brazil and Macao, we found that SRL constructs, such as self-efficacy and metacognitive strategies, are associated with programming achievement. Using a programming-specific instrument to measure SRL strategies enables us to corroborate that high-achieving students highly report the use of metacognition. Moreover, the results presented by the systematic literature review supports the relevance of SRL for programming education. Still, results should be carefully analysed. As correlation should not be confused with causality, the results from this review should be accompanied by other evidence to understand the influence of SRL on programming education. In conjunction with the results from our previous systematic literature review on SRL interventions [5], [6], there is an indication that SRL is a relevant phenomenon for helping programming learning. Replications of this study are needed to identify if our findings are only associated with our sample or generalise to the programming education community.

The aforementioned results provides an opportunity to investigate this phenomenon further using experimental studies, particularly how the interventions can change students' behaviour and support programming learning.

References

[1] Y. Qian and J. Lehman, "Students' misconceptions and other difficulties in introductory programming: A literature review," *ACM Transactions on Computing Education (TOCE)*, vol. 18, no. 1, pp. 1–24, 2017.

[2] A. V. Robins, "Novice programmers and introductory programming," *The Cambridge Handbook of Computing Education Research, Cambridge Handbooks in Psychology*, pp. 327–376, 2019.

[3] B. J. Zimmerman and D. H. Schunk, "Self-regulated learning and performance," *Handbook of self-regulation of learning and performance*, pp. 1–12, 2011.

[4] E. Panadero, "A review of self-regulated learning: Six models and four directions for research," *Frontiers in psychology*, vol. 8, p. 422, 2017.

[5] L. Silva, A. J. Mendes, A. Gomes, and G. F. Cavalcanti de Macêdo, "Regulation of learning interventions in programming education: A systematic literature review and guideline proposition," in *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, 2021, pp. 647–653.

[6] J. Prather, B. A. Becker, M. Craig, P. Denny, D. Loksa, and L. Margulieux, "What do we think we think we are doing? metacognition and self-regulation in programming," in *Proceedings of the 2020 ACM Conference on International Computing Education Research*, 2020, pp. 2–13.

[7] S. Bergin, R. Reilly, and D. Traynor, "Examining the role of self-regulated learning on introductory programming performance," in *Proceedings of the first international workshop on Computing education research*, 2005, pp. 81–86.

[8] R. Garcia, K. Falkner, and R. Vivian, "Systematic literature review: Self-regulated learning strategies using e-learning tools for computer science," *Computers & Education*, vol. 123, pp. 150–163, 2018.

[9] C. Watson and F. W. Li, "Failure rates in introductory programming revisited," in *Proceedings of the 2014 conference on Innovation & technology in computer science education*, 2014, pp. 39–44.

[10] J. Broadbent and W. L. Poon, "Self-regulated learning strategies & academic achievement in online higher education learning environments: A systematic review," *The Internet and Higher Education*, vol. 27, pp. 1–13, 2015.

[11] P. A. Alexander, D. L. Dinsmore, M. M. Parkinson, and F. I. Winters, "Self-regulated learning in academic domains," *Handbook of self-regulation of learning and performance*, pp. 393–407, 2011.

[12] E. G. Poitras and S. P. Lajoie, "A domain-specific account of self-regulated learning: The cognitive and metacognitive activities involved in learning through historical inquiry," *Metacognition and Learning*, vol. 8, no. 3, pp. 213–234, 2013.

[13] L. Zheng, "The effectiveness of self-regulated learning scaffolds on academic performance in computer-based learning environments: A meta-analysis," *Asia Pacific Education Review*, vol. 17, no. 2, pp. 187–202, 2016.

[14] K. Falkner, C. Szabo, R. Vivian, and N. Falkner, "Evolution of software development strategies," in *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, vol. 2. IEEE, 2015, pp. 243–252.

[15] D. Loksa and A. J. Ko, "The role of self-regulation in programming problem solving process and success," in *Proceedings of the 2016 ACM conference on international computing education research*, 2016, pp. 83–91.

[16] M. Dresel and M. Haugwitz, "The relationship between cognitive abilities and self-regulated learning: evidence for interactions with academic self-concept and gender," *High ability studies*, vol. 16, no. 02, pp. 201–218, 2006.

[17] P. R. Pintrich and E. V. De Groot, "Motivational and self-regulated learning components of classroom academic performance." *Journal of educational psychology*, vol. 82, no. 1, p. 33, 1990.

[18] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering–a systematic literature review," *Information and software technology*, vol. 51, no. 1, pp. 7–15, 2009.

[19] O. H. Lu, J. C. Huang, A. Y. Huang, and S. J. Yang, "Applying learning analytics for improving students engagement and learning outcomes in an moocs enabled collaborative programming course," *Interactive Learning Environments*, vol. 25, no. 2, pp. 220–234, 2017.

[20] D. G. Bonett and T. A. Wright, "Sample size requirements for estimating pearson, kendall and spearman correlations," *Psychometrika*, vol. 65, no. 1, pp. 23–28, 2000.

[21] H. Tian and F. Wu, "An empirical study of factors influencing undergraduates' programming learning engagement in blended learning environment," in *2018 Seventh International Conference of Educational Innovation through Technology (EITT)*. IEEE, 2018, pp. 6–10.

[22] H. Cigdem, "How does self-regulation affect computer-programming achievement in a blended context?" *Contemporary Educational Technology*, vol. 6, no. 1, pp. 19–37, 2015.

[23] F.-Y. Kuo, W.-H. Wu, and C. S. Lin, "An investigation of self-regulatory mechanisms in learning to program visual basic," *Journal of Educational Computing Research*, vol. 49, no. 2, pp. 225–247, 2013.

[24] J. Campbell, D. Horton, and M. Craig, "Factors for success in online cs1," in *Proceedings of the 2016 acm conference on innovation and technology in computer science education*, 2016, pp. 320–325.

[25] A. Lishinski, A. Yadav, J. Good, and R. Enbody, "Learning to program: Gender differences and interactive effects of students' motivation, goals, and self-efficacy on performance," in *Proceedings of the 2016 ACM Conference on International Computing Education Research*, 2016, pp. 211–220.

[26] C. Huang, "Achievement goals and self-efficacy: A meta-analysis," *Educational Research Review*, vol. 19, pp. 119–137, 2016.

[27] J. W. Kotrlik, H. A. Williams, and M. K. Jabor, "Reporting and interpreting effect size in quantitative agricultural education research." *Journal of Agricultural Education*, vol. 52, no. 1, pp. 132–142, 2011.

[28] C.-Y. Tsai, "Improving students' understanding of basic programming concepts through visual programming language: The role of self-efficacy," *Computers in Human Behavior*, vol. 95, pp. 224–232, 2019.

[29] R. M. Ryan and E. L. Deci, "Intrinsic and extrinsic motivation from a self-determination theory perspective: Definitions, theory, practices, and future directions," *Contemporary Educational Psychology*, vol. 61, p. 101860, 2020.

[30] A. Gomes, W. Ke, C.-T. Lam, M. J. Marcelino, and A. Mendes, "Student motivation towards learning to program," in *2018 IEEE Frontiers in Education Conference (FIE)*. IEEE, 2018, pp. 1–8.

[31] J. M. Bartels and J. J. Ryan, "Fear of failure and achievement goals: A canonical analysis," *Journal of Instructional Psychology*, vol. 40, no. 2, p. 42, 2013.

[32] D. Loksa, B. Xie, H. Kwik, and A. J. Ko, "Investigating novices' in situ reflections on their programming process," in *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 2020, pp. 149–155.

[33] C.-Y. Law, J. Grundy, A. Cain, R. Vasa, and A. Cummaudo, "User perceptions of using an open learner model visualisation tool for facilitating self-regulated learning," in *Proceedings of the Nineteenth Australasian Computing Education Conference*, 2017, pp. 55–64.

[34] K. Mangaroska, K. Sharma, M. Giannakos, H. Trætteberg, and P. Dillenbourg, "Gaze insights into debugging behavior using learner-centred analysis," in *Proceedings of the 8th international conference on learning analytics and knowledge*, 2018, pp. 350–359.

[35] J. Guerra, R. Hosseini, S. Somyurek, and P. Brusilovsky, "An intelligent interface for learning content: Combining an open learner model and social comparison to support self-regulated learning and engagement," in *Proceedings of the 21st international conference on intelligent user interfaces*, 2016, pp. 152–163.

[36] J. Prather, R. Pettit, K. McMurry, A. Peters, J. Homer, and M. Cohen, "Metacognitive difficulties faced by novice programmers in automated assessment tools," in *Proceedings of the 2018 ACM Conference on International Computing Education Research*, 2018, pp. 41–50.

[37] J. Robertson, "The educational affordances of blogs for self-directed learning," *Computers & Education*, vol. 57, no. 2, pp. 1628–1644, 2011.

[38] A. P. Ambrosio and S. W. Martins, "Study attitudes and behaviour of cs1 students-two realities." in *Psychology of Programming Interest Group (PPIG)*. Citeseer, 2011, p. 18.

[39] P. Virtanen and A. Nevgi, "Disciplinary and gender differences among higher education students in self-regulated learning strategies," *Educational psychology*, vol. 30, no. 3, pp. 323–347, 2010.