"An Observational Study of Undergraduate Teaching Assistants' Use of Subgoal Learning Integrated in an Introductory Programming Course"

Goletti, Olivier ; Mens, Kim ; Hermans, Felienne

ABSTRACT

In this paper we study an approach to support undergraduate teaching assistants - UTAs - in their teaching with more pedagogical content knowledge. This paper builds upon prior work on the integration of explicit strategies to observe and analyse how UTAs make actual use of subgoal learning (SL) as explicit instructional strategy in their daily teaching practice. From empirical evidence, we extract instructional design advice on how to integrate different subgoal learning strategies throughout a CS1 course. Our study 1) focuses on the effects of SL integration in UTA practice, 2) explores fidelity of their implementation of SL and 3) studies the different ways UTAs use the strategies. We use observations and surveys to analyse UTAs interventions during their lab sessions, and cross these observations results with UTAs' self-reported data through two surveys. Our main results are that SL integration through exercises is a major trigger of strategy use. Training and follow-up of the UTAs have impact, since UTAs feeling familiar with subgoal learning correlates positively with their successful observed use of subgoal learning. More frequent and stronger uses of the strategy were observed by UTAs self-reporting more frequent uses. And finally, UTAs express that subgoals are best suited for introducing concepts and especially the more structured ones.

CITE THIS VERSION

Goletti, Olivier ; Mens, Kim ; Hermans, Felienne ; et. al. *An Observational Study of Undergraduate Teaching Assistants' Use of Subgoal Learning Integrated in an Introductory Programming Course.*ACM SIGPLAN International Symposium on SPLASH-E (Pasadena, CA, USA, du 23/10/2024 au 25/10/2024). In: Felienne Hermans,Rose Bohrer, *SPLASH-E '24: Proceedings of the 2024 ACM SIGPLAN International Symposium on SPLASH-E*, ACM : NY, USA2024 http://hdl.handle.net/2078.1/292688 -- DOI : 10.1145/3689493.3689986

An Observational Study of Undergraduate Teaching Assistants' Use of Subgoal Learning Integrated in an Introductory Programming Course

Olivier Goletti
Université Catholique de Louvain
Louvain-la-Neuve, Belgium
olivier.goletti@uclouvain.be

Kim Mens
Université Catholique de Louvain
Louvain-la-Neuve, Belgium
kim.mens@uclouvain.be

Felienne Hermans
Vrije Universiteit Amsterdam
Amsterdam, Netherlands
f.f.j.hermans@vu.nl

Abstract

In this paper we study an approach to support undergraduate teaching assistants - UTAs - in their teaching with more pedagogical content knowledge. This paper builds upon prior work on the integration of explicit strategies to observe and analyse how UTAs make actual use of subgoal learning (SL) as explicit instructional strategy in their daily teaching practice. From empirical evidence, we extract instructional design advice on how to integrate different subgoal learning strategies throughout a CS1 course.

Our study 1) focuses on the effects of SL integration in UTA practice, 2) explores fidelity of their implementation of SL and 3) studies the different ways UTAs use the strategies. We use observations and surveys to analyse UTAs interventions during their lab sessions, and cross these observations results with UTAs' self-reported data through two surveys.

Our main results are that SL integration through exercises is a major trigger of strategy use. Training and follow-up of the UTAs have impact, since UTAs feeling familiar with subgoal learning correlates positively with their successful observed use of subgoal learning. More frequent and stronger uses of the strategy were observed by UTAs self-reporting more frequent uses. And finally, UTAs express that subgoals are best suited for introducing concepts and especially the more structured ones.

CCS Concepts: • **Social and professional topics** → **Computer science education**.

Keywords: Subgoal learning, Programming course, UTA training, CS1, observational study

SPLASH-E '24, October 24, 2024, Pasadena, CA, USA

ACM ISBN 979-8-4007-1216-6/24/10
https://doi.org/10.1145/3689493.3689986

ACM Reference Format:
Olivier Goletti, Kim Mens, and Felienne Hermans. 2024. An Observational Study of Undergraduate Teaching Assistants' Use of Subgoal Learning Integrated in an Introductory Programming Course. In *Proceedings of the 2024 ACM SIGPLAN International Symposium on SPLASH-E (SPLASH-E '24), October 24, 2024, Pasadena, CA, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3689493.3689986

1 Introduction

Introductory programming courses (CS1) are notoriously challenging as demonstrated by their high failure rates. But it does not have to be that way. Since a lot of CS departments use (under)graduate teaching assistants - (U)TAs - as teaching staff for their courses [8, 11], we decided to assist and train them with more pedagogical content knowledge (*"subject matter knowledge* for teaching*"* [28]).

Based on educational theories of *cognitive load theory* [31, 32] and *learning transfer* [4, 34] and their impact on instruction design [16, 33], we decided to integrate *subgoal learning* (SL) [5, 18] in a CS1 programming course that uses a problem-based learning methodology. The SL explicit instructional strategy consists in teaching explicitly the generic solving steps (the subgoals for a concept) an expert goes through when solving a problem using a specific concept. The steps are then labelled and the labels can be explicitly used by the teaching staff. The goal being to reduce cognitive load for students and fostering learning transfer. In this study, we integrated SL throughout the course by integrating these labels in the different resources of the course: course's slides, exercises' statement, UTAs training and material.

Extending previous research [14, 15] on the integration of explicit strategies and following design research principles in education [2], our goal is to bring research to practice [27] and to observe and analyse how 21 UTAs make actual use of subgoal learning as explicit instructional strategy in their daily teaching practice. We extract empirical evidence for instructional design advice on the integration of subgoal learning throughout a CS1 course. Our study answers the following research questions:

RQ1 What are the effects of subgoal learning integration on the UTAs' uses of the strategy?

RQ2 How do UTAs use the subgoal learning strategy according to our fidelity of implementation criteria?
RQ3 Can we classify UTAs' usage of the strategy in categories according to their observed usage and self-reported opinion on their usage of subgoal learning?

We will also discuss briefly what is the level of students' awareness of the strategy.

2 Background and Related Work

Subgoal learning was introduced by Catrambone [5] with the idea to foster learning transfer. It combines the worked-example effect and the annotation of steps or subgoals with labels, in order to highlight the generic structure of a problem solving procedure. Worked examples have been studied and shown to help students learn programming [9, 22]. Highlighting and explicitly teaching the steps of recurring patterns in the resolution of similar problems is not new in programming education [7, 30]. Margulieux et al. [20] combined the idea of worked examples with labeling the steps behind common code structures, proposing subgoal labelled worked examples (SLWEs) in computing education. Margulieux et al. showed that subgoal learning helps *"students learn more effectively without increasing the amount of time students take to learn"* [17]. They argued that subgoal learning benefits come from the combined effect of worked examples and reduced cognitive load on the learners and that it aids the development of mental models which helps for learning transfer. Atkinson et al. also found that *"sequentially-presented examples with clearly isolated subgoals produce better conceptual performance"* [1].

The selection of this strategy as an instruction intervention for training UTAs in order to foster learning transfer and reduce cognitive load [24, 33] has been discussed in previous works [14, 15].

UTAs in Computer Science. This study aims to bring research to practice by bringing evidence-based instructional strategies to Undergraduate Teaching Assistants (UTAs)' mentoring. UTAs or undergraduate tutors are heavily used in computer science courses [21] because they allow to break down large students cohorts into more efficient classrooms settings, allowing these highly demanded courses to scale [11]. In their literature review, Mirza et al. mention benefits to UTAs and to students. But being an UTA does not come without challenges [26]. Riese et al. mention student-focused challenges like teaching students with different profiles, but also what best practices to use to teach CS content properly and threats like time constraints [26].

Regarding training, some universities have dedicated courses to provide UTAs with a.o. pedagogical knowledge on learning theories, assessment and classroom management [6, 25]. While our UTAs also follow such a course [29], the aim of this research is to bring pedagogical content knowledge specific to a CS1 course to foster learning transfer for students.

Design Research in Education. The goal of design research is to produce actionable knowledge to achieve an educational goal through the design of instruction [2]. It is used to guide, test and refine educational practices, to fill a research gap in the practical design of instruction in a real-life classroom setting. This process often goes through multiple iterations and is guided by design principles. Typically, the output of a design research will be a list of design principles on characteristics and methodological aspects to include in the instruction supported by both theoretical and empirical arguments [2]. Examples in computing education research exist for introducing programming in primary schools [12] or the design of teacher ebooks integrating worked examples [10]. We were also inspired to use this approach by Margulieux et al.: *"A key idea in this paper is that instructional design matters. The two groups did not differ in the content of the instruction but in the design of that material (e.g., whether subgoals were made explicit)."* [19]

Fidelity of Implementation. Since we are making sense of how UTAs can integrate a new instructional strategy in their practice, we also wanted to have a tool for this. Research looks at their appropriation through two constructs: the *adaptation* that teachers will inevitably make and the *fidelity of implementation* of that strategy defined as the degree to which procedures are implemented as planned [23]. In an effectiveness study, where we look at the integration of an evidence-based strategy in a classroom setup, adaptation is inevitable. This study aims at documenting more accurately the impact of design choices to integrate subgoal learning by also looking at fidelity of implementation to how accurately and when the strategy is used [3] by UTAs (**RQ2**).

Prior Work. The current study uses a design research process [2] to build upon prior work that aimed at integrating explicit instructional strategies in a CS1 course to foster learning transfer and decrease cognitive load for students learning how to program [14, 15]. While previous work trained a small proportion of volunteer UTAs with multiple strategies,our study is focusing on subgoal learning (SL), integrates the strategy globally throughout the course and involves all 25 UTAs. **RQ1** will study the effects of this integration on the UTAs' uses of the strategy.The most mentioned issue by UTAs in focus groups and follow-up discussions in previous work was that, for them, to use SL during their lab sessions took too much time [15]. Indeed, they had to introduce the strategy to the students for the first time and then present a specific, provided, worked example and the corresponding subgoals and labels before being able to reference them in their instruction. UTAs argued that they would benefit from an introduction of the subgoals and labels directly in the course material or by the teachers. We also observed that for the other strategy used in that study (i.e. explicit tracing), an important trigger was to have explicit prompts in students' exercises statements [15]. These results led us to fully integrate SL throughout the CS1 course material.

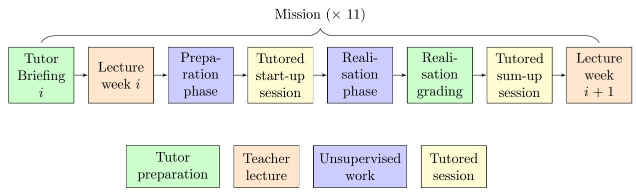


Figure 1. The different moments that structure a mission of the CS1 course (adapted from Goletti et al. [15]).

3 Study Design and Methodology

This section describes the methodology followed to integrate subgoal learning in the course and to collect and analyse data during the study. Fig. 2 presents a comprehensive timeline of the different moments in the semester when training and data collection were done.

We used a mixed-method approach in our study. We used qualitative analysis through both the coding of observations and surveys. To answer **RQ3** we wanted to be able to cross the results of the observations of the 21 UTAs made by our 6 observers, the UTAs' self-reported impression on their appropriation of the strategy and the students' perspective.

In Fig. 2, for each observed week, an id of the type "wXmY" is used. It stands for "week number X of the semester and mission number Y of the course"[1]. Apart from the training week (w3m2), three other weeks were observed, each of which with its associated course topics and concepts:

- **w5m4** on list traversals and functions;
- **w7m6** on file manipulation and exceptions;
- **w12-13m11** on the linked list data structure.

3.1 CS1 Course Structure

While the structure of the CS1 course has already been presented in detail in a previous publication [15], we still provide a quick summary here. The course consists of 11 missions spread over 12 weeks. Each mission introduces different programming concepts. For example, mission 10 introduces object-oriented programming. Fig. 1 presents the typical organisation of a mission.

3.2 Subgoal Learning Integration

In our current study, we identified key resources of the course that could be impacted by the integration of Subgoal Learning (SL) and designed a fading approach for the integration of subgoals and labels in the course, in accordance with CLT inspired instructional principles. The present study makes use of our own subgoals and labels developed and adapted to the concepts and language (Python) of our own CS1 course [13]. Table 1 presents for each course moment as presented in Fig. 1, what staff is involved, what resources mobilised and how SL was integrated.

The subgoals used in our study are specific to the concepts seen in the course and adapted to Python. They were written following a task analysis procedure [13]. The concepts covered are: assignment, conditionals, while loop, for loop and sequence traversal, functions, file read, file write, dictionary update, adding a node to a linked list and removing a node from a linked list.

Table 1. Resources used in the course per moment and how they were adapted to integrate subgoal learning. (T = Teachers; TA = Teaching Assistants; UTA = Undergraduate Teaching Assistants; S = Students)

moment	staff	target	resource	SL
Tutor Briefing i	T + TA	UTA		oral
Lecture week i	T	S	introduction slides	SBS
Preparation phase	-	S	exercises syllabus	prompts
Tutored start-up session	UTA	S	exercises syllabus	RQ
Realisation phase	-	S	exercises syllabus	-
Realisation grading	UTA		correction guide	SLFS
Tutored sum-up session	UTA	S	correction guide	RQ
Lecture week $i + 1$	T	S	restructuring slides	SLFS

We distinguish two different types of subgoal labelling utilisation. (1) *Step-By-Step Subgoal labeled worked examples* (SBS) are worked examples that use each label one by one to illustrate at each step how the generic corresponding subgoal for that concept is used in a concrete example. The integration of such SBSs is mainly done by teachers during their lectures, but a UTA using the labels and solving procedure to provide students with an exercise solution could also use SBSs. In order to integrate SBSs in the course, we had to modify teachers' slide decks and this took some work back and forth with teachers. (2) *Subgoal Labeled Final Solutions* (SLFS) would be used after the labels had already been introduced once. The aim is to show final solution or solved examples without necessarily going through the whole solving process but by annotating what code pertains to what label. Typically, those would be shown in the teachers restructuring slides to remind students of the labels or the result of an UTA annotating code on the blackboard during a lab session.

[1]Typically, Y = X - 1, except for the last mission wich was spread over two semester weeks.

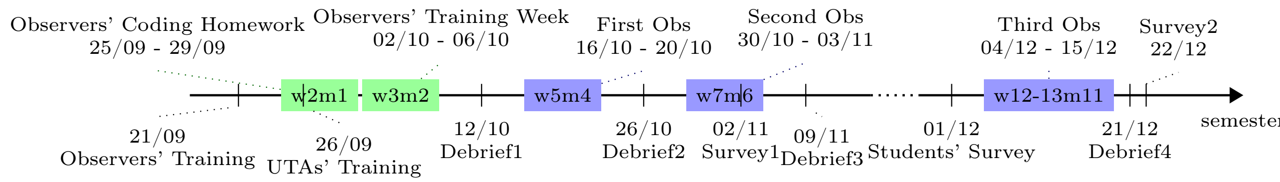


Figure 2. Timeline of the semester with observers' training weeks in green and actual observations in purple.

The integration of SL in the course consisted in the modification of the slides with SBSs and SLFSs, as shown in Fig. 3. This allowed for the introduction and repetition of the labels for the students in a passive way. Since it was also new for the teachers, some details on the extended subgoals, not just the labels, were also added in their presenter notes. Since we knew that prompts in students' exercises statements were also triggers for strategy use, we also added the labels in the first exercises making use of the targeted concepts. The labels are linked each time to a complete SLFS listing, called the catalogue, where the subgoals, labels and annotated examples (SLWEs) are gathered per concept. This full catalogue is available to the students as an appendix of the exercises syllabus. Prompts are a list of the labels or simplified subgoals with a link to this catalogue.

The last integration step relates to the UTAs. Next to the training of UTAs, we add reminders in the correction guide for UTAs that they will use for the *"Realisation Grading"* [2]. This helps them to know where and when new concepts are used and which labels to use. This information is also discussed during the weekly *"Tutor Briefing"* meetings with the UTAs.

3.3 UTAs' Training and Follow up

In our course design, the subgoal learning integration is systemic throughout the whole course, its material as well as its teaching methodology. All UTAs followed a mandatory training before the beginning of the course, consisting of theoretical background on the SL strategy, a description and demonstration of when and how to use the strategy, recording extracts from previous observations of good use and misuse and previous results, like the main triggers for strategy use [15]. Follow up sessions for the UTAs occurred during the weekly *"Tutor Briefing"* meetings. Questions were asked weekly on UTAs' strategy uses, feedback and rich discussions occurred between the UTAs and with the staff. Subgoals and labels corresponding to the weekly concepts were discussed during those meetings and strategy use was encouraged.

[2]I.e., the grading of and feedback by UTAs on a weekly assignment to be made by the students on the course topic of that week.

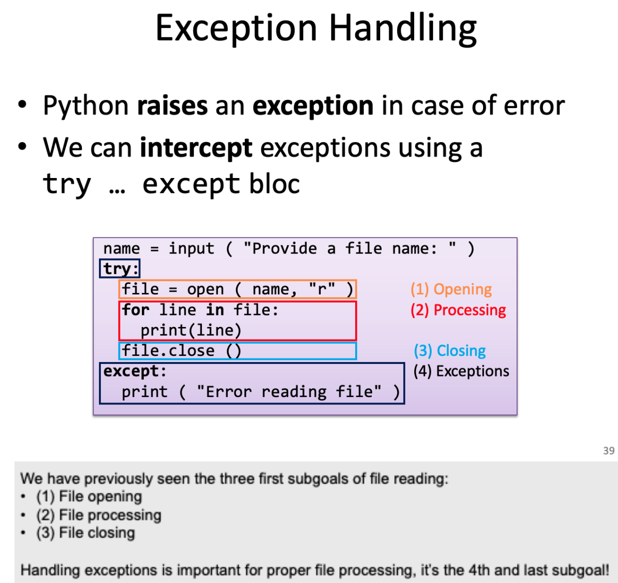


Figure 3. Modified slide with the code annotated with (French) subgoals and the presenter notes for the teacher. The material has been translated from French for readability.

3.4 Fidelity of Implementation and Deductive Codes

The criteria used to measure fidelity of implementation were adapted from previous work [15] to take into account the integration of the presentation of the different subgoals in the course. Subgoals and labels are now introduced directly during the lectures by the teachers, so that tutors don't have to introduce them during their one hour lab sessions. The criteria, derived from literature, adapted and refined for this study are:

SL1 The strategy or the catalogue is mentioned;
SL2 The tutor solves a similar problem in a worked example;
SL3a The tutor reminds the students about one of the subgoals of a concept;
SL3al The tutor reminds the students about one of the subgoals of a concept with mention of the label;
SL3b The tutor reminds the students about *all* the subgoals of a concept;

SL3bl The tutor reminds the students about *all* the subgoals of a concept with mention of the labels;

SL4 The solving procedure of a concept is used to solve a practice exercise with the students.

During the coding, we noticed some observers made a difference in coding when UTAs mentioned one or all the subgoals of a concept and with or without the provided labels. Since this was an interesting distinction that brought nuance to strategy usage, it led us to separate SL3 from the last paper across these two dimensions. SL3a and SL3b are with one or multiple subgoals but without mention of the labels, whereas in SL3al and SL3bl the labels are explicitly mentioned. So in the end we split SL3 into four sub-criteria that are mutually exclusive. This mention of the subgoals can either be done orally or by code annotation.

Based on this distinction, we will also introduce the notion of *"stronger"* use of the strategy when either two or more criteria are being used at the same time, or when SL3 is used with the labels being mentioned explicitly (SL3*l).

3.5 Observations

Since the intervention of this study involved a full integration of an instructional strategy at the scale of a whole CS1 course (~600 students), it involved 21 UTAs and 22 different lab sessions. The 11 missions of the course (2 lab sessions per week) are organised following five different series, dependent on the students' program, each with their own schedule. Two series for a total of 358 engineering students, two series for a total of 169 computer science students, and one series for 69 mathematicians and students with other backgrounds. Together, the series amounted to 27 different classrooms of ~24 students, each with their own (U)TA or in total 594 sessions to be observe. Among these 27 classrooms, 25 are tutored by UTAs (the other two by doctoral students, TAs), but only 21 of them agreed to participate in the study (ie. being observed and recorded during the whole semester).

In order to observe enough different UTAs on different sessions, 6 master students were recruited as *observers*. Five of them were paid as part of a student's job and the sixth one participated in the scope of his master thesis.

The observers were trained in the SL strategy with the same material used for tutors but also on what to observe and how to report it. They were tasked with the recording of the sessions and (pre)coding of those videos. Since the observers are CS master students and not qualitative researchers, they needed a training for their observation and coding tasks. Also, they were asked to do deductive coding based on the fidelity of implementation criteria of Sec. 3.4. We assigned them to code three recording extracts from past observations. Their coding was then discussed and corrected. Moreover, a first round of observation in classrooms was dedicated as a training week, so they could test their recording setup (their smartphone), and the time it took them to code. Most of them took notes during the lab sessions, then later watched again the recording to refine their coding. Both the recording and coding of each session was then uploaded to a shared repository according to a simple naming convention `obs/YYYYMMDD-HHHH-room`. The first author was able to download, watch and control the observation process.

Each observed week, observers were randomly assigned to three to four sessions. For example, for w7m6, one observer was assigned two *"Tutored start-up sessions"* (noted session w7m6-1): those of tutors T15 and T18 and two *"Tutored sum-up sessions"* (noted session w7m6-2): those of T14 and T13.

Each observer did between 8 and 11 observations spread over the three selected weeks as seen in Fig. 2. For each session they observed, they had to record the session and code the observed strategy uses by the UTA. A line in the coding file consists of a timestamp, some notes on the context, an eventual quote and the code itself. For each strategy use, they were asked to also note the trigger. Their coding was then reviewed by the first author who coded entirely three sessions of each observer. The SL3 codes had to be corrected and some strategy uses that were not generic enough had to be removed, some repetitions of codes for a same strategy use also had to be removed. Nevertheless, globally the observers coded accurately and thanks to their notes and quote fields, all corrections could be applied to the entire coding.

3.6 Surveys

UTAs' Surveys. To answer **RQ3**, we crossed the observations data with self-reported data from UTAs through surveys. At two points in the semester, UTAs were asked to fill a survey on their usage of the strategy. They were asked Likert-scaled questions on familiarity, frequency of use and opinion on the pedagogical value of the strategy. They were also provided with more open questions on their strategy usage. For the first survey, we asked for their agreement level on the following items:

S1Q1 I feel confident about giving my lab sessions;

S1Q2 I can see how to use the subgoals in practice in my lab sessions;

S1Q3 I had time to prepare my sessions this week;

S1Q4 I was influenced by the subgoals to prepare my sessions this week;

S1Q5 I used the subgoals during my sessions this week;

S1Q6 I thought about using the subgoals during my sessions this week but didn't actually do it.

For the second survey, the same items as the five last ones of the first survey were proposed but with a more global timeline (i.e. "my sessions this week" was replaced by "my sessions"). Moreover, seven new opinion items were added in the form of "I think the subgoals are useful":

S2Q7 for all students;

S2Q8 for stronger students;

S2Q9 for weaker students;

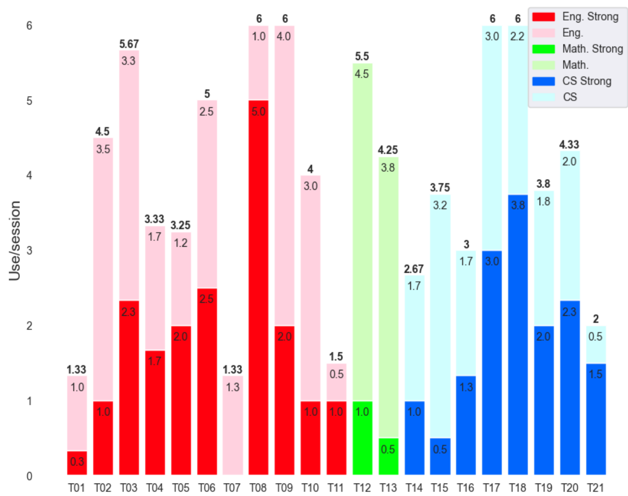


Figure 4. Normalised number of use of the SL strategy per session for each UTA. The darker bottom part of each bar corresponds to *"stronger"* uses. In red: the engineering program, in green: the math program, in blue: the CS program.

S2Q10 for all concepts;
S2Q11 for some concepts;
S2Q12 more for introducing concepts;
S2Q13 more for restructuring concepts.

Students' Survey. One survey was also passed to the students of the course. It mainly consisted of Likert-scaled questions on their awareness, familiarity and perception of the subgoal learning strategy and its integration in the course.

4 Results

4.1 Observations

In the end, 74 observations were made, 12 of which during w2m1 for training purposes. Each lab session is supposed to last one hour but the UTAs have sometimes room for more, so we ended up coding more than 62 hours of video.

Strategy Uses. Fig. 4 presents the normalised number of uses and *"stronger"* uses per session of the strategy for each UTA[3]. With a mean of 3.92 observed uses of the strategy per session (see Table. 4), we can answer **RQ1**: the integration of subgoal learning throughout the course pushed UTAs to effectively use the strategy. We can also see that the vast majority (16 out of the 21 studied UTAs) use on average 3 times or more the strategy per session. The rest of this section will answer **RQ2** on the observed fidelity criteria.

From Table 2 we can see that the most used criterion is SL3 (214 times in total), i.e. mentioning one or more subgoals with or without especially mentioning the labels. This occurs naturally in a UTAs' mentoring when reexplaining a concept.

[3]UTAs are referred to as TXX (T form tutor, XX is a id like 01) throughout the Results Section.

Table 2. Number of occurrences of each observed combination of criteria (with SL3 sub-criteria summed for readability).

criteria	SL1	SL2	SL3	SL4	n_occ
			×		138
		×	×		44
		×			20
			×	×	13
	×		×		9
	×				8
	×		×	×	8
		×		×	1
		×	×	×	1
	×	×	×		1
total	26	67	214	23	243
%	11	28	88	9	100

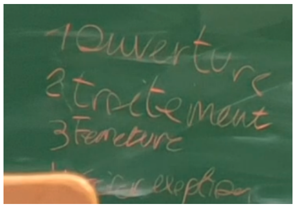
Only generic subgoals explanations were coded as such, not explanations linked to the specific context of an exercise.

From Table 2 we also see that SL3 was mostly used alone. Drilling down in detail[4], it was flagged as criterion SL3a (53 times), SL3al (38×), SL3bl (26×) and finally SL3b (21×) or in total 138 times or more than half of the total 243 strategy uses. The strategy was mostly used to explain a single subgoal at a time (53+38=91 with only one subgoal vs. 26+21=47 with multiple subgoals) and the labels are explicitly mentioned 38+26=64 times or nearly for half of the SL3 single uses.

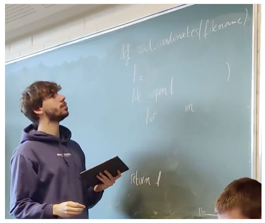
Illustrated further by Table 2, when SL3 is used with other criteria, it is mostly (44×) with SL2 for detailing the steps in a worked example. This corresponds to an UTA demonstrating how to use a specific concept while mentioning the subgoals they want to highlight. This is done either orally or by annotating code on the blackboard. An example of SL3bl with blackboard annotations is provided in Fig. 5a. For oral SL3 strategy usage, the triggers were mostly the exercise prompts or the correction of an exercise. SL3bl was observed 17× with SL4 (among which 8× also with SL1), those are the *"strongest"* uses of the strategy we can hope for. Again, adaptations of the strategy usage were also observed, for example six UTAs would make the students recite the subgoals for a concept (like writing a function or opening a file). One UTA also proposed a skeleton of a solution code based on the subgoals for helping the students to visualise this generic template (see Fig. 5c).

The second most used criteria is SL2, i.e. providing students with a worked example. This was mainly used for demonstrating the use of a concept, at the start of a session or to illustrate an alternative solution. SL2 was used 20× alone as a criterion, and 44× with SL3 as mentioned above. An example of SL2 and SL3 is provided in Fig. 5d.

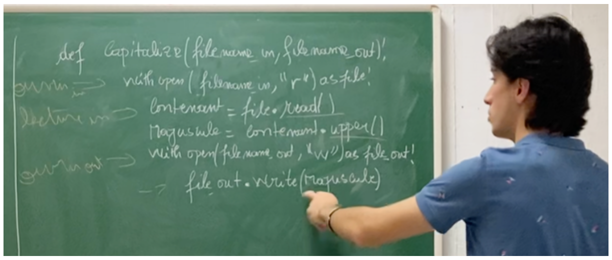
[4]For conciseness and readability these details were dropped from Table 2.



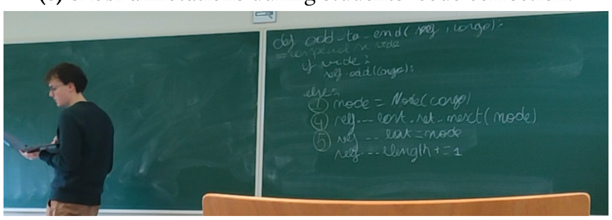
(a) SL3bl example.



(b) SL3bl (oral) with SL4 example with code skeleton.



(c) SL3bl annotations during students' code correction.



(d) SL3bl with SL2 example.

Figure 5. Some strategy use examples from the recordings.

SL4 was only observed together with other criteria, and mainly with SL3bl which makes total sense since it represents a reuse of the subgoals to solve another similar problem with the tutors. SL1, mentioning the catalogue, was observed 26× in total and only 8× alone. An example of SL4 used with SL3bl is an UTA annotating or mentioning the subgoals labels and using them to solve a new example, what we called earlier an SBS for step-by-step use of the strategy. An example is provided in Fig. 5b.

Apart from these criteria uses, we can also see in Table 4 that w7m6 accounted for the most strategy uses. That week concepts were file manipulations. If we look at the upper part of the table, both w7m6-1 and w12-13m11-1 have the most strategy uses per session. We can also see that the start-up sessions gathered significantly more strategy uses than sum-up sessions, meaning that UTAs use the strategy more often when introducing concepts.

There seemed to be little or no impact on strategy usage with respect to the student audience, i.e., either engineering students or computer science students.[5]

[5]Since the Math. audience was only observed six times it is hard to draw conclusions from that.

Table 3. Number of observed triggers by source.

source	trigger	code	n
UTA	before an exercise	t1	40
	as a correction check-list	t2	39
	before a session	t3	21
	to show a better solution	t4	13
	as an exercise for students	t5	8
	sub-total		**121**
Exercise	prompts	e1	63
	sub-total		**63**
Student	question from a student	s1	29
	to help student solve exercise	s2	18
	misunderstanding	s3	7
	bug in a student's code	s4	5
	sub-total		**59**
Total			**243**

Triggers. Table 3 summarises what triggered UTAs to use the strategy. First, we notice that UTAs remain the main trigger for strategy use and mainly for reintroducing a concept at the beginning of a session of an exercise, or as a check-list to restructure the concept after an answer has been given. Then, we can see that the exercise prompts we added accounted for one out of four strategy. When students are the triggers, it is mainly when they ask a question or when they need to be guided to solve an exercise, that UTAs will use the strategy. The effect of exercise prompts weight also in favor of SL integration.

4.2 UTAs' Surveys

Of the overall 25 UTAs of the course, 21 agreed to take part in the research meaning being observed, recorded and letting us use their data for this research by answering surveys. Among those, 12 answered the first survey and 11 the second one. The first survey was sent to the UTAs in the middle of the semester after all concepts had not yet been seen in the course (see Fig. 2 for details). The second survey took place after the end of the course. UTAs' answers will help us answer **RQ3** on classification of UTAs' usage of the strategy.

Familiarity and Frequency. The results of the Likert-scaled questions on familiarity and frequency of use for the different concepts are shown in Fig. 7a, 7b, 7c and 7d.

The data is only partial since not all UTAs answered the surveys. Overall, we can say that the familiarity of the UTAs with the concepts is reported as high. There's more variation in understanding of file operations and advanced topics. Between the two surveys, most UTAs maintained or improved their familiarity with the different concepts. T02, T05 and T20 have a lower aggregated score but due to the newest, more advanced concepts.

Table 4. Number of times each fidelity of implementation criterion was observed per mission, session or program. * means all different values are considered, which makes the last line a aggregated total of all observed uses. The "**SL3 (Σ)**" column adds up the four SL3 sub-criteria. The last column gives a normalised strategy uses per session value.

mission	session	prog.	SL1	SL2	SL3a	SL3al	SL3b	SL3bl	SL3 (Σ)	SL4	uses	strong uses	n	uses/ session
w5m4	1	*	5	9	14	5	4	5	28	4	35	11	9	3.89
	2		2	16	10	5	6	7	28	3	34	15	11	3.09
	*		7	25	24	10	10	12	56	7	69	26	20	3.45
w7m6	1	*	4	10	16	13	6	24	49	6	63	31	10	6.30
	2		2	10	12	14	2	5	33	1	35	11	12	2.92
	*		6	20	28	27	8	29	92	7	98	42	22	4.45
w12-13m11	1	*	9	13	12	4	9	14	39	9	45	23	9	5.00
	2		4	9	8	7	7	5	19	0	31	11	11	2.82
	*		13	22	20	11	16	19	66	9	76	34	20	3.80
*	1	*	18	32	42	22	19	43	128	19	143	65	28	5.11
	2		8	35	30	26	15	17	88	4	100	37	34	2.94
*	*	Eng.	14	39	29	20	15	30	94	13	111	48	28	3.96
		CS	11	26	28	19	16	29	92	10	104	50	26	4.00
		Math.	1	2	15	9	3	1	28	0	28	4	6	4.67
*	*	*	26	67	72	48	34	60	214	23	243	102	62	3.92

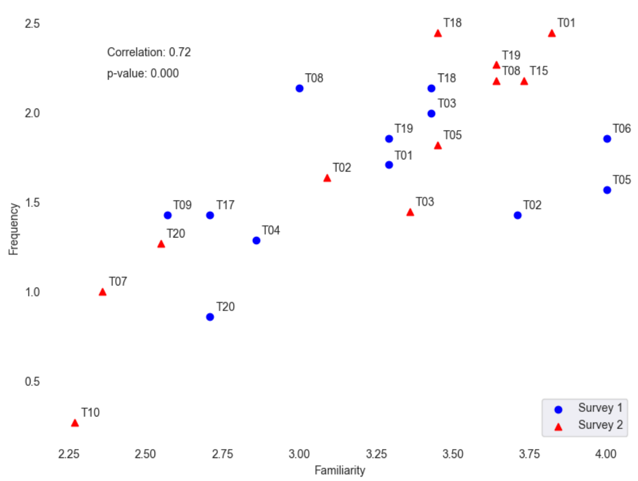
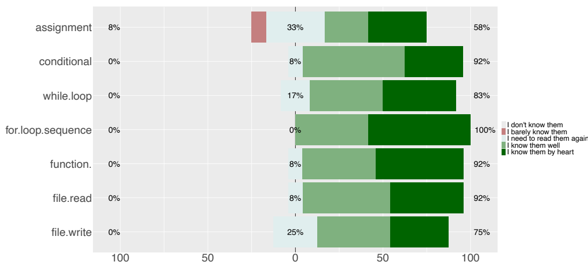


Figure 6. Correlation between UTAs' reported familiarity and frequency of use of the SL strategy.

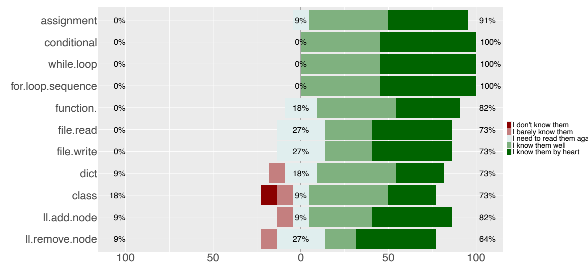
Regarding the frequency, there's a general trend towards increased frequency of use for most concepts among the UTAs who participated in both surveys. The best scoring concepts are loops, file operations and linked list operations.

We found that UTAs' self-reported familiarity correlates positively with their self-reported frequency (Pearson's correlation $r = 0.72, p = 0.00$) as presented in Fig. 6.

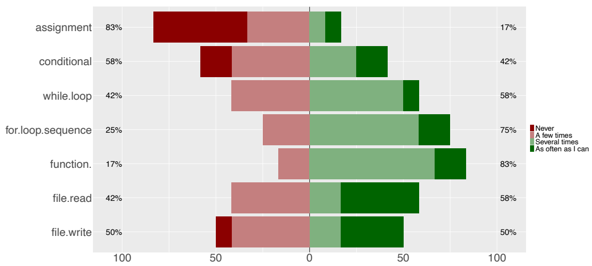
UTAs' Opinion of the Strategy. From the survey results presented in Fig. 7e and 7f, we can see that Q2 on UTAs vision on how to use subgoals got overall better results in the second survey, which is good news meaning that with time, UTAs felt they got better at applying the strategy. Q3's result on whether they had enough time to prepare their sessions are a bit worse but overall UTAs complained more about time during the sessions than the time to prepare. Q4, on the influence of subgoals on UTAs' preparation is a bit better in the second survey. This is, like Q2, interpreted by the amount of practice and experience UTAs have gained with using the strategy by the end of the course. In their open answers, the UTAs who used the labels specified they were useful to structure the session, prepared them to know how to answer students' questions or to help them remember the content of the course. Q5 on whether they declare using the subgoals during their sessions is roughly the same. Three out of four declared using the subgoals during their classroom sessions and the rest is neutral (neither agrees nor disagrees with the statement). T06 says on this account: *"In my opinion, I used the optimum number of steps; using more would have been counterproductive"*. Q6 asked about whether they at least *thought about* using the labels even if in the end they didn't; here also the results got better with two thirds declaring agreeing with this item. This shows at the same time more reflection on the UTAs part, but also that they either forgot or decided not using it during their sessions. Some UTAs said explicitly that they sometimes chose not to use the strategy: *"I used it less for simpler or mastered concepts, in order to not confuse students"* (T08). UTAs mentioned other obstacles too. Four mentioned not enough time during sessions and three



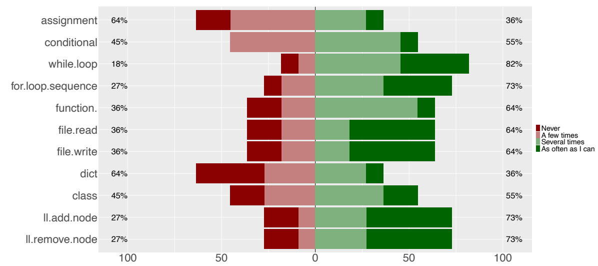
(a) Familiarity of use of concepts reported in first survey.



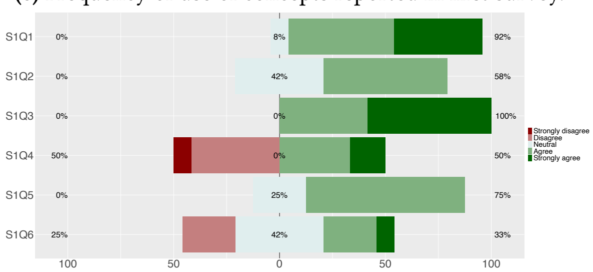
(b) Familiarity of use of concepts reported in second survey.



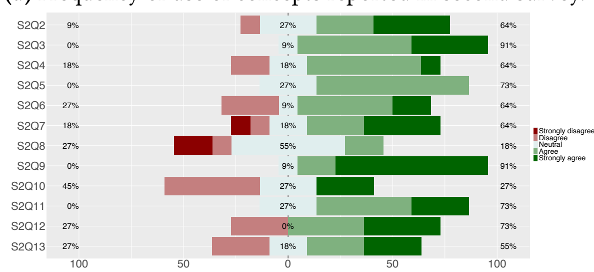
(c) Frequency of use of concepts reported in first survey.



(d) Frequency of use of concepts reported in second survey.



(e) Opinion on the strategy as reported in first survey.



(f) Opinion on the strategy as reported in second survey.

Figure 7. UTAs' aggregated scores on Likert-scaled questions on their familiarity, frequency and opinion on the use of SL on the two surveys.

mentioned that they themselves did not learn with them: *"I haven't learned to code with subgoals, I sometimes forget the point and realise it when a student gets stuck on something that shouldn't be too complicated"* (T05). T03 mentioned that a lot of exercises do not use all subgoals of a concept.

Questions S2Q7–13 interrogate the perceived usefulness of the strategy. S2Q7–9 concern the student audience the strategy helps. UTAs think the strategy is mainly useful for weaker students and one in four do not agree it helps stronger students. Answers to open questions refine this. While one in four totally agrees that the strategy if useful for all concepts, three out of four agree that it's at least useful for some concepts. Finally, UTAs think that SL is more suited to introduce concepts while half of them still think it can also be useful for restructuring concepts.

In the open questions, T03 suggested that *"the first exercises for each concept should nearly force students to use the subgoals or to add multiple choice questions on the subgoals"*, but that is something the teachers explicitly forbade. T18 also suggested adding subgoals for unit tests.

4.3 Crossing the Data Sources

UTAs' opinions on when to use the strategy corresponded to their actual use of the strategy which happened more often during introduction sessions than sum-up sessions. T01 said *"When students have trouble getting started ..., I show everyone what to do in general at the beginning (the steps to follow)."* which matches both triggers t3 and s1 from Table 3. T18 said something similar: *"To help students if the student cannot start, or it they start writing code directly, to highlight the subgoals (sometimes with the help of the classroom) in colour"*. UTAs also reported using the strategy more for some concepts, especially the more structured and advanced ones. This is in line with the higher use of the strategy during w7m6-1 (introduction to file operations) and w12-13m11-1 (introduction to linked list operations) as seen in Table 4. They agreed that the concept of file operations was more structured in the sense that the structure was more visible and therefore more easily mapped to named subgoals. UTAs also agreed they used the strategy more to illustrate a concept when used for the first time, to introduce it before a session or

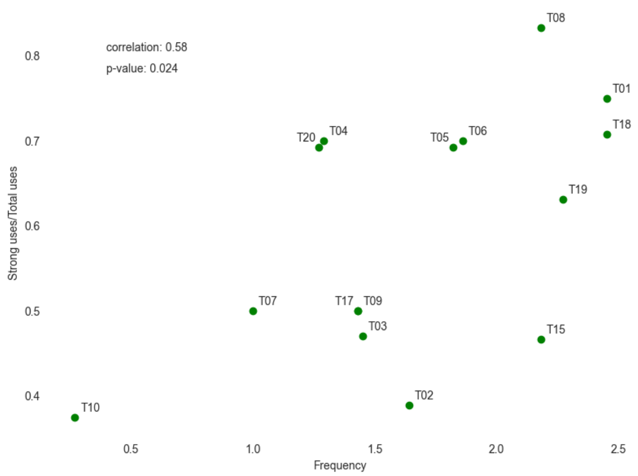


Figure 8. Correlation between UTAs' reported frequency of strategy use and their observed *"stronger"* use ratio.

before a specific exercise. This also aligns with the observed higher use for introducing concepts than during sum-up sessions (session 2 in Table 4).

We did find a significant moderately positive correlation (Pearson's correlation $r = 0.58, p = 0.024$) between UTAs self-reported frequency and the ratio of their observed number of *"stronger"* (i.e. at least two criteria or an explicit mention of the labels) uses to the total uses, see Fig. 8. We don't know yet if that correlation is just accidental or could be strengthened by giving UTAs even more guidance.

4.4 Students' Awareness of the Strategy

An optional survey was sent to the students before the last course mission. It was answered by 48 students only, which is less than 10% of all students. The survey consisted of eight questions on three themes. Overall, regarding awareness and frequency of mentioning the catalogue and the subgoals, a majority of the respondents seemed unaware of this strategy. Regarding the usefulness of the catalogue, even though the strategy seemed not to be seen that frequently by the students, most of them did consider it moderately to very helpful. Regarding the different concepts finally, it seems students think they know the first concepts treated by the subgoal labels well and the later ones a bit less.

4.5 Threats to Validity

While we ensured that UTAs were comfortable with the observations process by having a training week and letting them opt out, at least one tutor did stop participating during the process. Meanwhile, at least three UTAs feedback on their teaching practice during their lab session when they knew they had been recorded. This shows a real interest to become better tutors.

While we had discussions with the teachers of the course, training was deemed unnecessary for them and we did not check how they discussed the labels since they had been integrated in their slides.

Since there were 6 observers, newly trained for coding, the first author verified and harmonized the coding of the different observations. This process was eased by regular meetings with the observers.

5 Conclusion

Subgoal learning is a promising teaching strategy for CS1 programming courses to enhance students' learning experience and outcomes. Through a detailed analysis of a large-scale CS1 course in which we fully integrated this strategy, and studied its fidelity of implementation among undergraduate teaching assistants (UTAs), our study revealed key insights.

We answered **RQ1** by acknowledging the effect of SL integration throughout the course, in particular the effect of exercise prompts on UTAs' strategy uses. Training and follow-up sessions with UTAs played a crucial role in the successful deployment of this strategy. As the course progressed, UTAs reported increased familiarity and frequency of use of the labels for the different concepts taught. This may indicate that the subgoal learning method helped streamline their instructional process and improved their teaching practice. We also found a moderately positive correlation between their self-reported frequency and the ratio of their observed number of stronger use over their total strategy uses.

We use fidelity criteria to answer **RQ2** and introduced the notion of *"stronger"* use of the strategy. The mention of subgoals remained the most observed use while around 40% of the strategy uses have been classified as *"stronger"* uses which seems correlate with UTAs reported frequency of use.

Answering **RQ3** to classify UTAs use of the subgoal learning strategy, we observed it was used more by UTAs during sessions introducing new concepts than during sum-up sessions, and more for structured and advanced concepts than for simpler ones. This structured dimension of concepts that would better fit with the strategy is a lead for further research. The program of the students audience had little impact on UTAs use of the strategy.

Another working point is to make the strategy's use more visible to the students. Despite students' infrequent perception of the strategy's mention during lab sessions and lectures, a majority found the subgoals to be moderately to very helpful. This suggests that even limited exposure to subgoal learning can positively influence student understanding and retention of programming concepts.

Overall, the findings from this design research study provide interesting insights on how subgoal learning can and is used as a tool by UTAs for reducing cognitive load and promoting learning transfer in introductory programming courses.

References

[1] Robert K Atkinson and Sharon J Derry. 2013. Computer-Based Examples Designed to Encourage Optimal Example Processing: A Study Examining the Impact of Sequentially Presented, Subgoal-Oriented Worked Examples 1. In *International Conference of the Learning Sciences*. Psychology Press, 132–133.

[2] Arthur Bakker. 2018. *Design research in education: A practical guide for early career researchers*. Routledge (Taylor & Francis), London and New York.

[3] Maura Borrego, Stephanie Cutler, Michael Prince, Charles Henderson, and Jeffrey E. Froyd. 2013. Fidelity of Implementation of Research-Based Instructional Strategies (RBIS) in Engineering Science Courses. *Journal of Engineering Education* 102, 3 (2013), 394–425. https://doi.org/10.1002/jee.20020

[4] Danièle Bracke. 2004. *Un Modèle Fonctionnel Du Tranfert Pour l'éducation*. Presses Université Laval, 77–106.

[5] Richard Catrambone. 1998. The subgoal learning model: Creating better examples so that students can solve novel problems. *Journal of experimental psychology: General* 127, 4 (1998), 355.

[6] Holger Danielsiek, Jan Vahrenhold, Peter Hubwieser, Johannes Krugel, Johannes Magenheim, Laura Ohrndorf, Daniel Ossenschmidt, and Niclas Schaper. 2017. Undergraduate Teaching Assistants in Computer Science: Teaching-related Beliefs, Tasks, and Competences. In *2017 IEEE Global Engineering Education Conference (EDUCON)*. Ieee, Athens, Greece, 718–725. https://doi.org/10.1109/educon.2017.7942927

[7] Michael de Raadt, Mark Toleman, and Richard Watson. 2007. Incorporating programming strategies explicitly into curricula. In *Proceedings of the Seventh Baltic Sea Conference on Computing Education Research - Volume 88* (Koli National Park, Finland) *(Koli Calling '07)*. Australian Computer Society, Inc., AUS, 41–52.

[8] Paul E. Dickson, Toby Dragon, and Adam Lee. 2017. Using Undergraduate Teaching Assistants in Small Classes. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (Seattle, Washington, USA) *(SIGCSE '17)*. Association for Computing Machinery, New York, NY, USA, 165–170. https://doi.org/10.1145/3017680.3017725

[9] Barbara J. Ericson, Lauren E. Margulieux, and Jochen Rick. 2017. Solving Parsons Problems Versus Fixing and Writing Code. In *Proceedings of the 17th Koli Calling International Conference on Computing Education Research (Koli Calling '17)*. Acm, New York, NY, USA, 20–29. https://doi.org/10.1145/3141880.3141895

[10] Barbara J. Ericson, Kantwon Rogers, Miranda Parker, Briana Morrison, and Mark Guzdial. 2016. Identifying Design Principles for CS Teacher Ebooks Through Design-Based Research. In *Proceedings of the 2016 ACM Conference on International Computing Education Research (Icer '16)*. Acm, New York, NY, USA, 191–200. https://doi.org/10.1145/2960310.2960335

[11] Jeffrey Forbes, David J. Malan, Heather Pon-Barry, Stuart Reges, and Mehran Sahami. 2017. Scaling Introductory Courses Using Undergraduate Teaching Assistants. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (Seattle, Washington, USA) *(SIGCSE '17)*. Association for Computing Machinery, New York, NY, USA, 657–658. https://doi.org/10.1145/3017680.3017694

[12] Katharina Geldreich, Alexandra Simon, and Peter Hubwieser. 2019. A Design-Based Research Approach for Introducing Algorithmics and Programming to Bavarian Primary Schools. *MediaEducation: Journal for Theory and Practice of Media Education* (Feb. 2019), 53–75 Seiten. https://doi.org/10.21240/mpaed/33/2019.02.15.x

[13] Olivier Goletti, Florian De Pierpont, and Kim Mens. 2022. Création d'exemples résolus avec objectifs étiquetés pour l'apprentissage de la programmation avec Python. In *Didapro 9–DidaSTIC*.

[14] Olivier Goletti, Kim Mens, and Felienne Hermans. 2021. Tutors' Experiences in Using Explicit Strategies in a Problem-Based Learning Introductory Programming Course. In *ITiCSE '21*. ACM Press, Virtual Event, Germany, 7. https://doi.org/10.1145/3430665.3456348

[15] Olivier Goletti, Kim Mens, and Felienne Hermans. 2022. An Analysis of Tutors' Adoption of Explicit Instructional Strategies in an Introductory Programming Course. In *Proceedings of the 22nd Koli Calling International Conference on Computing Education Research*. 1–12.

[16] Paul A. Kirschner. 2002. Cognitive Load Theory: Implications of Cognitive Load Theory on the Design of Learning. *Learning and Instruction* 12, 1 (Feb. 2002), 1–10. https://doi.org/10.1016/s0959-4752(01)00014-7

[17] Lauren Margulieux, Richard Catrambone, and Mark Guzdial. 2013. Subgoal Labeled Worked Examples Improve K-12 Teacher Performance in Computer Programming Training. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, Vol. 35.

[18] Lauren E. Margulieux, Richard Catrambone, and Mark Guzdial. 2016. Employing subgoals in computer programming education. *Computer Science Education* 26, 1 (Jan. 2016), 44–67. https://doi.org/10.1080/08993408.2016.1144429 Publisher: Routledge.

[19] Lauren E. Margulieux, Mark Guzdial, and Richard Catrambone. 2012. Subgoal-Labeled Instructional Material Improves Performance and Transfer in Learning to Develop Mobile Applications. In *Proceedings of the Ninth Annual International Conference on International Computing Education Research (Icer '12)*. Association for Computing Machinery, New York, NY, USA, 71–78. https://doi.org/10.1145/2361276.2361291

[20] Lauren E. Margulieux, Briana B. Morrison, and Adrienne Decker. 2019. Design and Pilot Testing of Subgoal Labeled Worked Examples for Five Core Concepts in CS1. In *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education - ITiCSE '19*. ACM Press, Aberdeen, Scotland Uk, 548–554. https://doi.org/10.1145/3304221.3319756

[21] Diba Mirza, Phillip T. Conrad, Christian Lloyd, Ziad Matni, and Arthur Gatin. 2019. Undergraduate Teaching Assistants in Computer Science: A Systematic Literature Review. In *Proceedings of the 2019 ACM Conference on International Computing Education Research* (Toronto ON, Canada) *(ICER '19)*. Association for Computing Machinery, New York, NY, USA, 31–40. https://doi.org/10.1145/3291279.3339422

[22] Briana B. Morrison, Lauren E. Margulieux, and Mark Guzdial. 2015. Subgoals, Context, and Worked Examples in Learning Computing Problem Solving. In *Proceedings of the Eleventh Annual International Conference on International Computing Education Research* (Omaha, Nebraska, USA) *(ICER '15)*. Association for Computing Machinery, New York, NY, USA, 21–29. https://doi.org/10.1145/2787622.2787733

[23] Carol L. O'Donnell. 2008. Defining, conceptualizing, and measuring fidelity of implementation and its relationship to outcomes in K–12 curriculum intervention research. *Review of educational research* 78, 1 (2008), 33–84. Publisher: Sage Publications.

[24] Fred Paas, Alexander Renkl, and John Sweller. 2003. Cognitive load theory and instructional design: Recent developments. *Educational psychologist* 38, 1 (2003), 1–4.

[25] Emma Riese and Viggo Kann. 2022. Training Teaching Assistants by Offering an Introductory Course. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education - Volume 1 (Sigcse 2022, Vol. 1)*. Association for Computing Machinery, New York, NY, USA, 745–751. https://doi.org/10.1145/3478431.3499270

[26] Emma Riese, Madeleine Lorås, Martin Ukrop, and Tomáš Effenberger. 2021. Challenges Faced by Teaching Assistants in Computer Science Education Across Europe. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1* (Virtual Event, Germany) *(ITiCSE '21)*. Association for Computing Machinery, New York, NY, USA, 547–553. https://doi.org/10.1145/3430665.3456304

[27] Sue Sentance. 2021. Teaching computing in school: is K-12 research reaching classroom practice?. In *Proceedings of the 21st Koli Calling International Conference on Computing Education Research* (Joensuu, Finland) *(Koli Calling '21)*. Association for Computing Machinery, New York, NY, USA, Article 1, 2 pages. https://doi.org/10.1145/3488042.3491040

[28] Lee S Shulman. 1986. Those who understand: Knowledge growth in teaching. *Educational researcher* 15, 2 (1986), 4–14.

[29] Piotr Sobieski, Delphine Ducarme, and Vincent Wertz. 2017. Renforcer l'analyse réflexive des tuteurs en formation. In *Actes - Questions de pédagogie dans l'enseignement supérieur, QPES*.

[30] Elliot Soloway. 1986. Learning to Program= Learning to Construct Mechanisms and Explanations. *Commun. ACM* 29, 9 (1986), 850–858.

[31] John Sweller. 1988. Cognitive load during problem solving: Effects on learning. *Cognitive science* 12, 2 (1988), 257–285.

[32] John Sweller, Paul Ayres, and Slava Kalyuga. 2011. *Cognitive Load Theory, Volume 1 of Explorations in the Learning Sciences, Instructional Systems and Performance Technologies*. Springer, New York.

[33] John Sweller, Jeroen JG van Merriënboer, and Fred Paas. 2019. Cognitive Architecture and Instructional Design: 20 Years Later. *Educational Psychology Review* 31 (2019), 1–32. Publisher: Springer.

[34] Jacques Tardif. 1999. *Le Transfert Des Apprentissages*. Editions logiques.

Received 2024-07-08; accepted 2024-08-08