Students' Help-Seeking Patterns in Subgoal Learning Environments for Programming

Kento KOIKE[a*], Rwitajit MAJUMDAR[b], Takahito TOMOTO[c], Tomoya HORIGUCHI[d] & Tsukasa HIRASHIMA[e]
[a]*Faculty of Engineering, Tokyo University of Science, Japan*
[b]*Research and Educational Institute for Semiconductors and Informatics, Kumamoto University, Japan*
[c]*Faculty of Innovative Information Science, Chiba Institute of Technology, Japan*
[d]*Graduate School of Maritime Sciences, Kobe University, Japan*
[e]*Graduate School of Advanced Science and Engineering, Hiroshima University, Japan*
*kento@koike.app

Abstract: Subgoal learning facilitates knowledge transfer by structuring learning around meaningful goals. While such environments offer feedback to guide learners, how students seek help using these tools—and how such behaviors impact learning—remains underexplored. This study investigates help-seeking patterns in a block-based programming ITS with multi-level feedback. We analyzed log data from 42 engineering students using KMeans clustering and identified 6 distinct profiles.

Keywords: Help-seeking, subgoal learning, programming education, clustering, behavioral analysis, intelligent tutoring systems, learning analytics

1. Introduction

Subgoal learning facilitates knowledge transfer by breaking complex tasks into meaningful steps (Catrambone & Holyoak, 1990). In programming education, it fosters hierarchical thinking and supports understanding of code logic (Koike et al., 2020, 2024; Margulieux & Catrambone, 2019). Learning systems often provide feedback tools such as hints and behavior visualizations, enabling students to reflect on their reasoning by comparing the expected and actual outcomes. Despite these tools' potential, little is known about how students seek help in subgoal-based learning environments. Help-seeking likely varies by strategy, prior knowledge, and problem-solving style, thereby affecting learning outcomes (Wiggins et al., 2021).

Therefore, this study explores help-seeking behaviors in Compogram (Koike et al., 2020, 2024), a block-based programming environment that integrates hierarchical subgoals and multi-level feedback, including behavior visualization. It addresses the following questions:

- RQ1: What help-seeking patterns emerge among students?
- RQ2: How are these patterns associated with learning outcomes?

2. Learning Environment and Data Collection

Compogram (Figure 1) is a subgoal-based intelligent tutoring system designed to facilitate students' understanding of programming structures and problem-solving approaches (Koike et al., 2020). Among the four feedback mechanisms implemented in the system, only behavior visualization and hints were included in the current analysis. These were classified as optional feedback, as learners could voluntarily choose to use them. In contrast, correctness evaluation and review function were categorized as mandatory support, as they are either automatically triggered or required for progressing through tasks. As these mechanisms do not represent voluntary help-seeking behavior, they were excluded from the following analysis.

To investigate students' help-seeking patterns in Compogram (Koike et al., 2020), we analyzed behavioral log data collected from a prior pre-post design experiment (Koike et al., 2024). The experiment was conducted during a 105-minute online session in a Java programming course for third-year engineering students in Japan. Data from students who provided informed consent and had complete records were included in the analysis (N = 42).

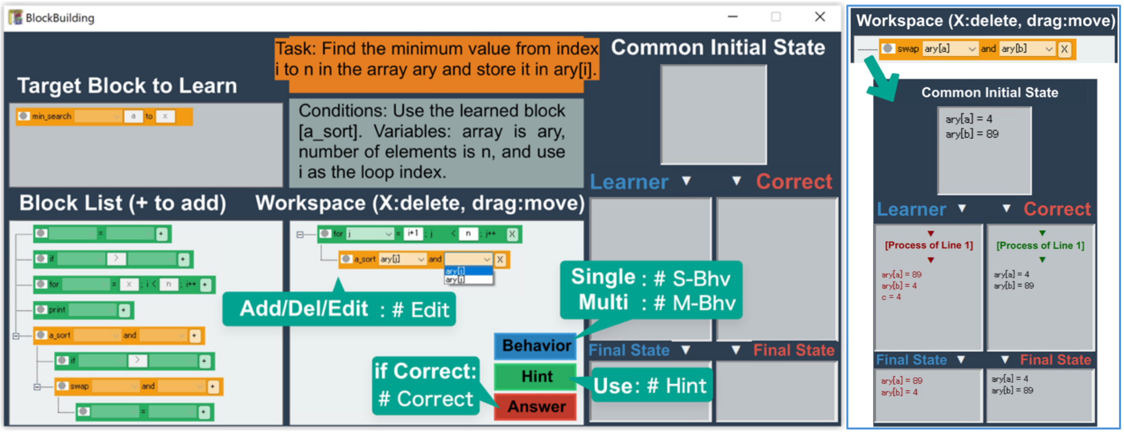

Figure 1. System interface and a visualization case of Compogram

3. Research Study

We conducted a clustering analysis using students' behavioral log data (N = 42). Redundant actions were removed during preprocessing. Unchanged workspace actions (e.g., dragging and returning a block) were excluded, and consecutive hint requests were counted once, as they provided the same information in an unchanged code state. For behavior visualizations, single executions were logged as S-Bhv. Since each run used randomized input, repeated executions were grouped and logged separately as M-Bhv. As a result, five key behavioral indicators were selected: frequency of single-step behavior visualizations (**# S-Bhv**), frequency of multi-step behavior visualizations (**# M-Bhv**), number of hint requests (**# Hint**), number of code edits (**# Edit**), and number of correct submissions during the learning phase (**# Correct**). We applied KMeans clustering to identify subgroups of students with similar behavioral profiles. The silhouette coefficient peaked at k = 6, suggesting that a six-cluster solution provided the most interpretable structure. After clustering, we profiled each cluster.

The clustering analysis produced six distinct behavioral profiles. Table 2 summarizes the cluster-wise behavioral indicators and shows behaviors per solving problem as **M/#C**. Each cluster was labeled for interpretability:

- Moderate Engagers (ME): Average use across feedback types with moderate correctness
- System Explorers (SE): High use of all system features, especially multi-behavior visualizations
- Reflective Achievers (RA): High correctness, extensive hint usage, minimal behavior visualization
- Minimal Participants (MP): Very low system engagement and no correct submissions
- Strategic Solvers (SS): High correctness with appropriate behavior feedback reliance rather than seeking hints
- Outlier Maximalist (OM): Extremely high interaction rates with low correctness (N = 1)

The largest groups were clusters ME (N = 14) and SS (N = 13). Cluster ME represented baseline learners, characterized by moderate engagement and moderate learning progress. Cluster SE showed high levels of engagement but only moderate learning progress. Notably, Clusters RA and SS demonstrated high progress despite different strategies—one with and the other without hint usage. Activities of clusters MP and OM were deemed either inappropriate (# Correct = 0, N = 2) or outliers (# S-Bhv (M/#C) = 28.67, N = 1).

Table 2. Summary of Six Clusters from KMeans

		# S-Bhv		# M-Bhv		# Hint		# Edit		# Correct	
		M (M/#C)	SD	M (M/#C)	SD	M (M/#C)	SD	M (M/#C)	SD	M	SD
ID	N										
ME	14	25.93 (3.16)	8.53	8.14 (0.99)	7.61	20.07 (2.44)	16.96	256.36 (31.23)	70.85	7.21	2.52
SE	6	44.67 (5.36)	19.55	43.33 (5.20)	15.04	45.17 (5.42)	26.75	420.67 (50.50)	71.89	7.33	1.97
RA	6	48.83 (3.45)	17.34	11.50 (0.81)	11.61	86.33 (6.09)	22.35	479.00 (33.80)	40.92	13.17	4.54
MP	2	0.00 (0.00)	0.00	0.50 (0.50)	0.71	0.00 (0.00)	0.00	27.00 (27.00)	24.04	0.00	0.00
SS	13	48.08 (3.83)	13.15	6.00 (0.48)	5.67	28.31 (2.26)	13.24	437.62 (34.90)	47.58	11.54	3.13
OM	1	172.00 (28.67)	-	8.00 (1.33)	-	46.00 (7.67)	-	550.00 (91.67)	-	5.00	-

Note: M/#C Indicates Mean divided **# Correct**+1

4. Discussion and Conclusions

This study identified six help-seeking profiles in a subgoal-based programming environment, highlighting that the focus and intentionality of feedback use shape how students engage with learning tools. Clusters RA and SS adopted distinct but effective strategies: RA used hints in a goal-directed, confirmatory way (Wiggins et al., 2021), while SS relied more on behavior visualization tools, forming a novel profile centered on reasoning grounded in program behavior.

In contrast, SE showed high tool use but lacked focus, resembling the “Help! Now what?” cluster (Wiggins et al., 2021) and suggesting patterns akin to wheel-spinning. ME showed moderate engagement but less progress. MP and OM reflected behavioral extremes that adaptive systems should detect and address.

Identifying the SS profile offers a novel perspective on visualization-based reasoning as an underrecognized but effective help-seeking strategy. Future studies should examine how these profiles relate to learning outcomes over time and explore the temporal dynamics of help-seeking behavior to inform adaptive support design.

Acknowledgments

This work was supported by JSPS KAKENHI (grant number JP25K21362).

References

Catrambone, R., & Holyoak, K. J. (1990). Learning subgoals and methods for solving probability problems. *Memory & Cognition*, *18*(6), 593–603.

Koike, K., Mogi, T., Tomoto, T., Horiguchi, T., & Hirashima, T. (2020). Compogram: Development and evaluation of ITS for organizing programming-knowledge by visualizing behavior. *HCI International 2020*, *151–162*, *12427 LNCS*.

Koike, K., Tomoto, T., Horiguchi, T., & Hirashima, T. (2024). *The effect of behavior visualization in supporting subgoal learning for programming [IEICE technical report]*, *124*(306), 27–34 (in Japanese).

Margulieux, L. E., & Catrambone, R. (2019). Finding the best types of guidance for constructing self-explanations of subgoals in programming. *Journal of the Learning Sciences*, *28*(1), 108–151.

Wiggins, J. B., Fahid, F. M., Emerson, A., Hinckle, M., Smith, A., Boyer, K. E., Mott, B., Wiebe, E., & Lester, J. (2021). Exploring Novice Programmers’ Hint Requests in an Intelligent Block-Based Coding Environment. *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, *52–58*.