University Students' Achievement Goals and Help-Seeking Strategies in an Intelligent Tutoring System

Bram Vaessen

Frans Prins

Johan Jeuring

Technical Report UU-CS-2013-019

Department of Information and Computing Sciences
Utrecht University, Utrecht, The Netherlands
www.cs.uu.nl

ISSN: 0924-3275

Department of Information and Computing Sciences
Utrecht University
P.O. Box 80.089
3508 TB Utrecht
The Netherlands

Abstract

Help seeking behavior in an intelligent tutoring system was analyzed to identify help seeking strategies, and it was investigated whether the use of these strategies could be predicted by achievement goal scores. Five strategies were identified, three of which were predicted by achievement goal scores. These strategies were labeled Little Help, Click through Help, Direct Solution, Step By Step, and Quick Solution. The Click Through Help strategy was predicted by mastery avoidance goals, the Direct Solution strategy was negatively predicted by mastery avoidance goals and positively predicted by performance avoidance goals, and the Quick Solution strategy was negatively predicted by performance approach goals.

Keywords: Intelligent tutoring systems; Interactive learning environments; Learning strategies

1. Introduction

Interactive learning environments (ILEs), which are used to train or teach novices, are becoming more widespread (Koedinger, Anderson, Hadley & Mark, 1997). Such an environment usually provides on-demand help functionality (Aleven, Stahl, Schworm, Fischer, & Wallace, 2003). In intelligent tutoring systems (ITSs), which are a type of ILE, the help functionality is usually a crucial part of the software as it can help to support a student individually by providing context sensitive hints and feedback. Compared to human tutoring, however, the software-tutor does not have access to verbal and non-verbal forms of communication (Wood & Wood, 1999). This means that ITSs place higher demands on a student's self-regulatory ability, because a student must decide for herself when she needs help and what kind of help she requires. It is, therefore, not surprising that students often do not use such help systems appropriately (Beal & Lee, 2008; Aleven et al., 2003).

To improve the value of ITSs by finding ways to stimulate more effective behavior, it is important to find out how a student uses these help systems and what factors influence this behavior (Karabenick, 2011). Currently, there is one model that describes desired help-seeking behavior within an ITS (Aleven, McLaren, Roll, Koedinger, 2006), but this model only captures desired versus undesired behavior, and its design was not based on empirical data. We took a more explorative approach to identify different help-seeking strategies within an ITS (based on Köck & Paramythis, 2011). Achievement goals are an important influence on help-seeking behavior in classroom situations, but have not been investigated well in ITSs (Aleven et al., 2003; Huet, Escribe, Dupyrat, & Sakdavong, 2011). We investigated the relation between the use of help-seeking strategies and students' achievement goals (Elliot & McGregor, 2001), We studied help-seeking strategies in an intelligent tutor for functional programming, in which a student incrementally solves functional programming problems. The tutor gives feedback, hints, and worked-out solutions based on annotated model solutions provided by teachers (Jeuring et al., 2012).

1.1 Intelligent Learning Environments and Intelligent Tutoring Systems

An interactive learning environments is a computer-based instructional system that offers a task environment and provides support to help novices learn skills or concepts involved in that task (Aleven et al., 2003). An ITS is a type of ILE that is designed for individual learning, and distinguishes itself from other types of ILEs by its ability to model the cognitive state of the user, which allows for context sensitive hints and feedback at all steps of a learning process (e.g. VanLehn et al. 2005; Graesser, Chipman, Haynes, & Olney, 2005). The advantage of using an ITS over other types of ILEs for *investigating* help-seeking behavior is that information about the cognitive state of the user is available, which may improve the interpretation of the observed help-seeking behavior. Although we studied the use of an ITS in this research, the approach for finding strategies can be used in any ILE that offers help functionality and is able to log user interactions.

1.2 Help Seeking in ITSs

Help seeking is conceptualized by Ames and Lau (1982) as "an achievement behavior involving the search for and employment of a strategy to obtain success." Ryan et al. identified the following steps in the help-seeking process: a) become aware of need of help, b) decide to seek help, c) identify potential helper(s), and d) use strategies to elicit help (Ryan, Pintrich, Midgley, 2001). In the first step, a student must become aware that the task is too difficult and that she is in need of help. In some situations this might be more difficult than it sounds; meta-cognitive skills such as self-monitoring and evaluation of own skills are required for this step. The next step is also crucial: even if a student acknowledges the need for help, she still has to decide whether she will seek help or not. A student might not seek help because she believes that, even with help, her basic competencies are not sufficient to solve the problem. Or she may have had prior experiences in which seeking help was not successful (Ames & Lau, 1982). Another reason for not seeking help is that a student may perceive seeking help as a threat to self-esteem or a threat to autonomy (Deci & Ryan, 1987;

Huet, Escribe, Dupyrat, & Sakdavong, 2011; Karabenick, 2004; Ryan, Pintrich & Midgley, 2001). This means she is either afraid of social embarrassment, or believes she will learn more if she tries without help. The third and fourth step refer to the social aspects of seeking help, and are not considered in this paper.

A differentiation is often made between *executive help seeking* and *instrumental help seeking* (Nelson-Le Gal, 1981, Karabenick, 2004). The goal of executive help seeking is to decrease the cost of completing a task by asking help from others, for example by asking for the answer to a problem. The goal of instrumental help seeking is to get the minimal assistance required to independently complete a task.

Help seeking in ITSs has been studied before (for a review, see Aleven et al, 2003), but not to the same extent as in classroom situations. It is not fully clear how much help seeking in ITSs differs from help seeking in classrooms, but it is often found that students avoid the help functionality, use it ineffectively, or abuse it. This is unfortunate, as appropriate help seeking in an ITS can improve learning. Several personal factors, but also the design of the help functionality, have been found to influence help-seeking behavior.

1.3 Measuring Help Seeking Behavior and Help Seeking Strategies in ITSs

1.3.1 Frequencies versus Sequences. One can measure help-seeking behavior in intelligent tutoring systems in several ways. One straightforward approach is to count the number of times a student asks for certain types of help while completing an exercise (Huet, Escribe, Dupeyrat, & Sakdavond, 2011; Wood & Wood, 1999). However, only considering the frequency of help seeking does not give enough information, because help seeking is a process, in which the necessity of help and the decision to seek help or not are important (Ryan et al. 2001). Imagine, for instance, an ITS that provides a user with two help buttons for each step; one button provides the user with a hint, the other button presents the answer to the current step. Two attempts at solving an exercise might consist of the following activities:

Student 1: ReqHint -> ReqAnswer -> TrySuccess -> TryFail -> TryFail -> TrySuccess.

Student 2: TryFail -> ReqHint -> TryFail -> ReqAnswer -> TrySuccess -> TrySuccess.

The frequencies of the activities of both students are the same, but their help-seeking behavior is rather different. Student 1 started the exercise by requesting a hint, as well as the answer for the first step, and requested no help after a failed attempt on the third step. In comparison, Student 2 only requested a hint after a failed attempt, and only requested an answer when the hint was not helpful. Thus, the sequence of activities can be more informative than the frequency.

Modeling sequences is not as straightforward as counting the frequencies of help requests. One way to quantify help-seeking behavior using sequences is to look for similarities in sequences, and to use these similarities to classify behavior. Köck and Paramythis (2011) modeled and clustered activity sequences to discover and classify help-seeking behavior. Activity sequences were modeled using Discrete Markov Models (DMM), and strategies were identified using a *k*-means clustering algorithm.

1.3.2 Discrete Markov Models and Help-Seeking Behavior. A DMM is a probabilistic model containing a finite number of N states, N x N transition probabilities, and N initial state probabilities. Each activity the student performs in an ITS is modeled as a state. Furthermore, for each pair of activities (states) a transition probability is calculated, based on sample data. For example, a transition probability of 0.3 for activity A to activity B signifies that if a student performed activity A, there is a 30% chance that the student will then perform activity B. This is valuable information because it can show, for example, how often a student requests an *answer* after a failed attempt compared to how often a student requests a *hint* after a failed attempt. All these transition probabilities taken together can be seen as a model for the student's interactions within an exercise. To only model help-seeking behavior, a subset of transition probabilities is selected. This subset (i.e. the combination of selected transition probabilities) is then considered a model of the help-seeking behavior of a student during an attempt on an exercise.

1.3.3 Clustering and interpreting help-seeking strategies. It is assumed that many help-seeking behaviors are similar, and that several typical *help seeking strategies* can be distinguished when students use the same electronic learning environment. To find these typical strategies, a clustering algorithm is used. Attempts on exercises with similar transition probabilities grouped into clusters. Because the transition probabilities in a cluster are similar, it is assumed the same help-seeking strategy was used in a cluster. The centers of a cluster, which are averaged transition probabilities, can be interpreted as a help-seeking strategy.

It is unclear how to determine the number of clusters that should be distinguished. Several methods have been proposed over the past few decades (Steinley, 2006; Tibshirani, Walther, Hastie, 2000), but none of these methods finds the 'correct' number of clusters in any given situation. Köck and Paramythis (2011) proposed a method that is specific to clustering strategies, providing a range of plausible number of clusters, but the final decision will still be taken by the researcher, based on theoretical arguments as well as the possibility to interpret clusters and the distribution of cases into clusters.

1.4 Achievement goals

Achievement goals are used to classify or measure motivation for competence-related behavior, for example in the 2x2 achievement goal framework from Elliot and McGregor (2001). This framework views achievement goals as the *purpose* of competence-related behavior (Maehr, 1989). It distinguishes two fundamental dimensions. The first dimension is called *definition*, which focuses on how a student defines (academic) competence. In this dimension, a differentiation is made between *mastery goals*, for which a student uses an absolute or intrapersonal standard for competence (i.e. a student considers herself competent when she masters some material, or if substantially improves herself), and *performance goals*, for which a student uses a normative standard (i.e. a student considers herself competent when her grade is higher than her friend's grade). The other dimension is called *valence,* in which a student who wants to attain a desired outcome (*approach goals*) is distinguished from a student who wants to avoid an undesirable outcome (*avoidance goals*).

The 2x2 framework results in four possible goals: *mastery approach*, *mastery avoidance*, *performance approach*, and *performance avoidance*. A student with mastery approach goals strives to be as competent as possible, while a student with mastery avoidance goals typically focuses on avoiding making mistakes, avoiding misunderstanding, or avoiding forgetting something she learned. In contrast, a student with performance approach goals strives to do well in comparison with other students, and a student with performance avoidance goals wants to prevent looking incompetent.

1.5 Help-seeking Behavior in ILEs and the Relation to Achievement Goals

Little research has been performed on the relation between achievement goals and help-seeking behavior in ITSs. From related research in classroom situations we do expect that some relations can be shown. In classroom situations it was found that a student with mastery goals uses instrumental help seeking more often, while a student with performance goals uses executive help seeking more often, or avoids seeking help (Arbreton, 1998; Karabenick, 2004; Bong, 2009). These results support the conceptualization of help seeking as a way of acquiring volitional control over the environment in the pursuit of learning goals (Kuhl, 1985), because the goal of help seeking seems to be in line with the achievement goal. Furthermore, it was found that a student with performance goals avoids seeking help because of a threat to self-esteem (Karabenick, 2004).

A difference between help seeking in class and help seeking in ILEs may be that there is less threat to self-esteem, because help requests are private and do not lead to social embarrassment (Aleven et al., 2006). This might imply that in an ILE, a student with performance goals does not request less help than a student with mastery approach goals. Huet, Escibe, Dupeyrat and Sakdavong (2011) still found a relation between performance goals and help avoidance in ILEs, which might indicate that the private image of competence can also threatened without social embarrassment. Furthermore, they found that mastery goals were related to a threat to autonomy, but not to the amount of help seeking.

1.6 Research Questions

We used the above research method to identify and describe help-seeking strategies in an ITS, and to relate them to achievement goals. We investigated the following research questions:

1. Which help-seeking strategies do students use in an ITS?
2. Can the use of these strategies be predicted by achievement goals?

Because we used an explorative approach for the first research question, no hypotheses were directly tested. Yet, based on the literature and previous findings, we expect to find at least one strategy that can be related to avoiding help, one executive help seeking strategy, and one instrumental help seeking strategy. As Huet et al. (2001), we expect that performance goals are related to strategies with help avoidance and executive help seeking strategies, and that mastery goals are related to instrumental help seeking strategies.

2. Material and methods

2.1 Participants

The participants were 210 students of the University of Utrecht who were taking an introduction class to functional programming (88% male, Mage = 21.2 years, age range = 18-49 years). From these participants, 58 individuals filled out a questionnaire about achievement goals but did not use the functional programming tutor, 21 individuals used the functional programming tutor but did not fill out the achievement goal questionnaire, and 45 individuals filled out the questionnaire and also used the functional programming tutor. This means that log files were available for 66 participants, and that 103 participants filled out the questionnaire.

2.2 Tutor

In the tutoring system, called the Functional Programming Tutor, students choose from 18 different assignments. The assignments involve developing a functional program in which lists of data elements are manipulated according to the specification of the exercise. The system supports several solutions for each problem, and each solution has automatically been split into refinement steps. The tutor analyzes both partial and complete student solutions and determines whether or not they fit one of the available solutions. It also recognizes code that is syntactically different from a supported solution, but functionally identical. A student can check her progress, which can be correct, contain a syntax error, or is not recognized as a step towards known solution. She can also ask for a hint on which step to take next in various levels of detail, or ask the tutor to complete the exercise. Access to these help functions is provided by textual buttons on a bar that is visible at all times during an exercise. The students can start and stop exercises at any point, and there are no penalties or limits on checking progress and seeking help. Figure 1 shows a screenshot of the tutor.

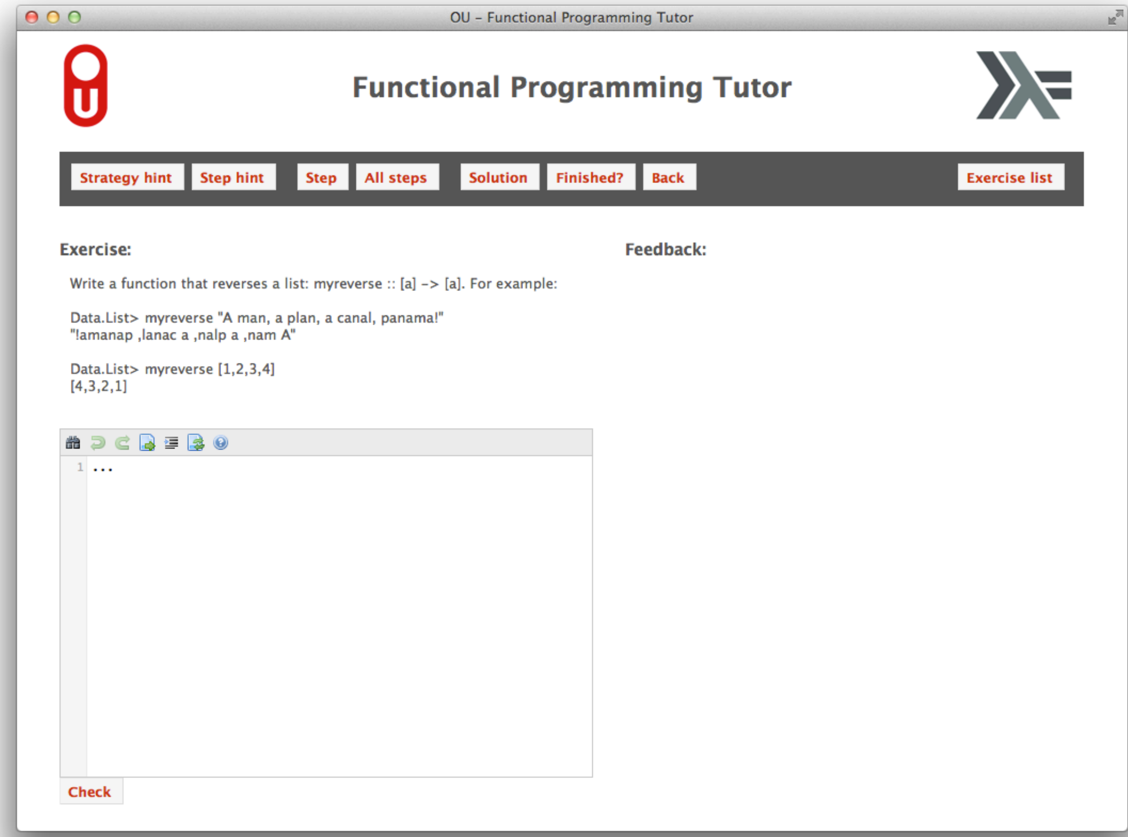

Figure 1. A screenshot of the tutor. The on-demand help buttons appear at the top. On the left, the exercise as well as a text editor containing the user input is available, and on the right the result of help requests is shown.

Depending on the choices of the user, each problem can be approached as a *problem-solving only* exercise (when no help is requested), a *solved-example problem* (when the solution is requested; Schworm & Renkl, 2006), or a *worked-out example* (when individual steps are consulted; Chi, Bassok, Lewis, Reimann, & Glaser, 1989). There are no instructional explanations in the tutor. The strategy and step hints are rather brief and do not explain why certain steps should be taken.

2.3 Instruments

2.3.1 Achievement Goals. The achievement goals were measured using the revised Achievement Goal Questionnaire (AGQ-revised; Elliot & Murayama, 2008), providing scores on the four achievement goals from the 2x2 achievement goal framework (Elliot and McGregor, 2001). The questionnaire consists of 12 questions, with 3 questions for each achievement goal. The four scales of achievement goals are *mastery approach*, *mastery avoidance*, *performance approach,* and *performance avoidance*. Sample questions for these scales are “my aim is to completely master the material presented in this class” for mastery approach, “my aim is to avoid learning less than I possibly could” for mastery avoidance, “my aim is to perform well relative to other students” and “my aim is to avoid doing worse than other students” for performance avoidance. Students responded on a scale from 1 (*strongly disagree*) to 5 (*strongly agree*). The four subscales have high internal consistency, with Cronbach’s α between .84 and .94 (Elliot & Murayama, 2008). In the current research the scales had an internal consistency between .76 and .90. The reliability analyses were based on all students who filled out the questionnaire (n = 103).

2.3.2 Help-seeking behavior. Before strategies can be distinguished, the help seeking behavior of students was modeled using the following steps. First, the log file with all user activities was divided into segments, each containing the list of activities in one attempt on an exercise. The first attempt on an exercise for each student was omitted from our analysis because the students were using the program for the first time, and we expected trial-and-error behavior in the first exercise (this resulted in excluding six students from our analysis, because they only tried one exercise). Then, for each segment a DMM was constructed. Table 1 shows the states that were used in the DMM. Some states directly refer to activities from the user, such as clicking a help button, while other states are activities combined with the response from the tutor.

Finally, the relevant transition probabilities were extracted from each DMM, which resulted in models of help-seeking behavior. To decide which transitions to select, we reasoned as follows: we were interested in how often and what type of help a user sought after she was told that her code was correct, or after she was told her code was incorrect. Furthermore, we wanted to know whether she changed her code in between help requests, and how often she asked for a more specific form of help directly after asking for a more general form of help. Finally, we were also interested in how a student used the help she received. This resulted in the selection of transitions as shown in Table 2.

2.4 Procedure

At the end of the third two-hour lecture of the course 'Introduction Function Programming' the students received an instruction on how to use the Functional Programming Tutor (Jeuring, Gerdes, & Heeren, 2011) in a ten-minute session. The teacher gave a general description of the tutor and its functionality, and showed on a large screen how to use the tutor. After the introduction, the students received a questionnaire about achievement goals and proceeded to the computer labs where they were able to work with the tutor for two hours. Before the lab session in which the students used the tutor they had taken two other lab sessions in which they got acquainted with a Haskell interpreter, and worked on simple programming exercises on numbers in Haskell.

Table 1

The states that were used in this study.

State	Meaning
ok[bde]	Student just started exercise, or clicked on the 'check' button and was informed that a correct step was made.
notok[b]	Student clicked on 'check' or 'finished' and was informed that the step was not recognized.
checkse[b]	Student clicked on 'check' and was told a syntax error was detected.
changecode[c]	Student changed the code before clicking on a new button.

strategyhint[a]	Student clicked on the 'strategy hint' button.
stephint[a]	Student clicked on the 'step hint' button.
step[a]	Student clicked on the 'step' or 'all steps' button.
copystep[c]	Student copied the code from the results of the 'step' button.
solution[a]	Students clicked on the 'solution' button.
copypartsol[c]	Student copied some part of the solution.
fincor[b]	Student clicks on the 'finished' button and was informed that the solution was correct.
stop[af]	Student quits exercise, or copied the full solution after clicking the 'solution' button and then quit the exercise.

[a]State is directly linked to an activity, such as pushing a button.
[b]State is combination of a activity and a response from the tutor.
[c]State is inferred by comparing the entered code to previous code or tutor response.
[d]No transition probability with this state was used.
[e]This state is also used as the starting state.

Table 2

The state transitions that were used for clustering in this study.

Transition probabilities		
ok -> strategyhint	changecode -> strategyhint	notok -> strategyhint
ok -> stephint	changecode -> stephint	notok -> stephint
ok -> step	changecode -> step	notok -> step
ok -> solution	changecode -> solution	notok -> solution
strategyhint -> stephint	stephint -> step	step -> solution
strategyhint -> step	stephint -> solution	step -> copystep
strategyhint -> solution		solution -> stop

2.5 Analyses

Help-seeking strategies. To find the different help seeking strategies that were used, the segments were clustered on the selected state transition probabilities, using the k-means clustering algorithm (Steinley, 2006). As the number of clusters is not known in advance, clustering was performed multiple times, each time with a different number of clusters, ranging from 2 to 20. Then, a more narrow range of a plausible number of clusters was determined with a method based on the one by Köck and Paramythis (2011). For each solution with *k* number of clusters the average cluster variance was calculated and normalized, as well as the average student entropies. The clustering solution for which the difference between the two measures was smallest was considered to have the best tradeoff between average variance within clusters and average student entropies. For both measures, a weight was assigned and varied, so this resulted in a range of plausible number of clusters.

To decide how many clusters to select, the distribution of segments in clusters, as well as the interpretability of the clusters was investigated. First, the amount of segments in clusters was evaluated. Solutions with clusters of very small size were not further evaluated, as those clusters are not very informative. Then, the cluster solution with the highest number of clusters that were all interpretable, was selected as final solution.

To interpret clusters, we paid special attention to variables that may help in distinguishing clusters, such as variables with means that deviated considerably from the grand average of that variable (large z-values), or deviated clearly from the means of other clusters. Variables with very high or low absolute means are interesting too, because they show transitions that are very likely or unlikely to happen from a certain state. Together, we used these variables as an indication of what the segments in clusters had in common, and how they differed from segments in other clusters, providing insight into the help-seeking strategy used in a cluster.

After selecting a cluster solution, it was checked whether or not the use of a strategy was related to individuals. This is important, because if the choice of strategy within an exercise does *not* depend on the person who performed the exercise, then there would be little use in investigating the relation between the use of a strategy and achievement goals. This assumption was tested using Fisher's exact test (Upton, 1992). We also test whether or not the choice of strategy is related to the exercise, which we expected; for instance because a more difficult exercise calls for a different strategy than an easy one. This indicates that the strategy identified behaves as expected. Although it might be interesting, we did not further investigate the exact nature of the relation between strategy use and exercise.

To measure whether or not a student used a strategy, we created a variable that denotes whether or not a student used this strategy at least once. We used four logistic regression analyses (n = 45) with the achievement goal scores as predictor, to find out if the use of strategies could be predicted with achievement goal scores. A *backward* entry method was used, with .1 as removal criteria, because this method fits well with a more explorative design.

3. Results

3.1 Descriptive Statistics for Help-seeking Behavior

The data yielded 271 segments, from 60 students. The number of segments for each student ranged from 1 to 16, with an average of 4.52 segments (SD=3.12). The number of unique exercises per students (some students performed the same exercise more than once) ranged from 1 to 11, with an average of 3.33 (SD=2.09).

For each segment we recorded whether or not a student asked for the solution, and whether or not the student finished the exercise without consulting the answer to the exercise, either in the current attempt, or a previous attempt on the same exercise (we refer to the latter as having *success* or being *successful*). Table 3 shows how many students tried an exercise and how many attempts were made in total on that exercise.. It also gives information about the average number of actions performed in an exercise, how often students asked for a solution, and how often they were successful.

Table 3

Number of students, total number of attempts, average number of actions, and success rates for all exercises.

Exercise	Students	Attempts	M_{actions} (SD)	Asked Solution	Success

ButLast	25	35	7.54 (4.66)	29%	20%
Compress	37	58	21.72 (25.06)	59%	10%
DropEvery	23	33	21.36 (26.39)	39%	3%
Dupli	21	26	11.19 (10.26)	15%	50%
ElementAt	27	38	19.68 (26.09)	39%	0%
Encode	7	9	21.00 (40.25)	67%	0%
MyConcat	8	8	12.38 (7.25)	38%	50%
MyLast	19	26	12.69 (7.40)	31%	8%
MyLength	14	18	21.61 (18.11)	78%	6%
MyReverse	9	10	22.40 (10.18)	20%	70%
Pack	1	1	7.00	100%	00%
Palindrome	4	4	21.25 (8.30)	0%	50%
Repli	3	3	9.33 (3.06)	67%	0%
Slice	2	2	46.50 (57.28)	100%	0%

Table 4 shows the descriptive statistics for the selected transition probabilities from the constructed DMMs, as well as three variables concerning the performance on the segments. The results show that the exercises differ considerably in number of attempts, length of attempts, and how often students asked for a solution or were successful.

Table 4
Descriptive Statistics for the Selected Transition Probabilities from the DMMs and for Performance Variables

Variable	Minimum	Maximum	*M*	*SD*
	Transition probabilities			
OkToStrategyhint	0.00	0.50	0.01	0.06
OkToStephint	0.00	0.90	0.10	0.20
OkToStep	0.00	0.80	0.07	0.16
OkToSolution	0.00	0.50	0.01	0.06
ChangecodeToStrategyhint	0.00	1.00	0.08	0.24
ChangecodeToStephint	0.00	1.00	0.03	0.13
ChangecodeToStep	0.00	1.00	0.05	0.16
ChangecodeToSolution	0.00	1.00	0.09	0.26

NotokToStrategyhint	0.00	1.00	0.05	0.18
NotokToStephint	0.00	1.00	0.06	0.19
NotokToStep	0.00	1.00	0.06	0.19
NotokToSolution	0.00	1.00	0.05	0.17
StrategyhintToStephint	0.00	1.00	0.06	0.20
StrategyhintToStep	0.00	1.00	0.03	0.15
StrategyhintToSolution	0.00	1.00	0.03	0.17
StephintToStep	0.00	1.00	0.12	0.28
StephintToSolution	0.00	1.00	0.02	0.10
StepToSolution	0.00	1.00	0.05	0.17
StepToCopystep	0.00	1.00	0.08	0.20
SolutionToStop	0.00	1.00	0.24	0.42
Performance variables				
ActionCount	3.00	128.00	17.3875	21.18
AskedSolution			42%	
Success			15%	

Note. N = 271. *ActionCount* is the number of actions within a segment. *AskedSolution* means the user has asked for the solution in the exercise.

3.2 Identified Help-seeking Strategies. The normalized average student entropies and the normalized average variance for 2 to 20 cluster solutions were calculated, and the weights attached to these metrics were varied between 1 and 3 (see Köck & Paramythis, 2011). This resulted in in between two and six plausible clusters. Figure 2 shows how the segments were assigned to clusters and what happened with the segments from the clusters in a solution with one more cluster.

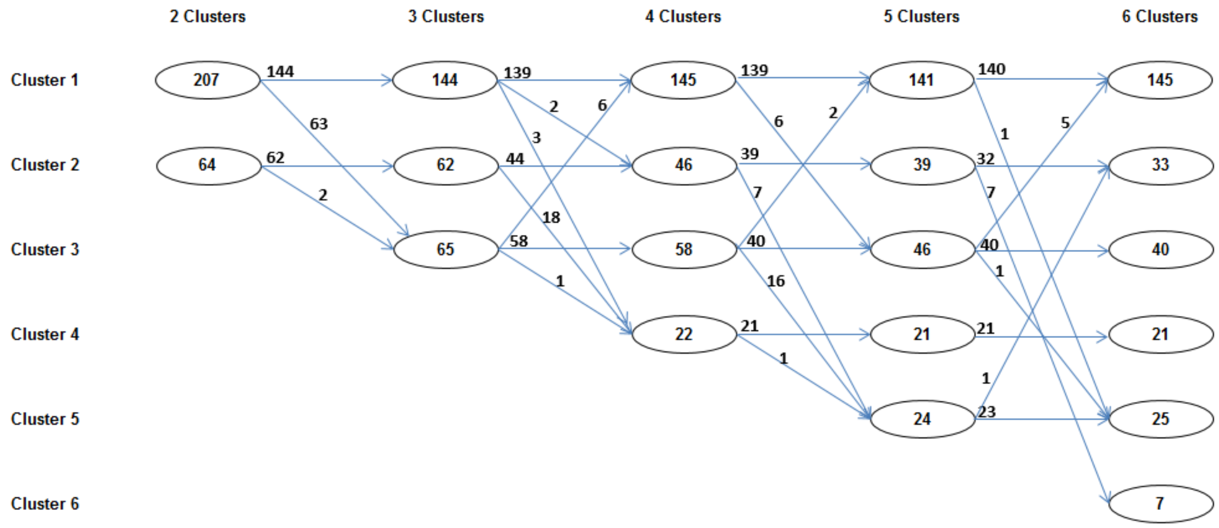

Figure 2. Number of segments per cluster for cluster solutions with two to six clusters. The numbers inside the ellipses denote the number of segments that were assigned to a cluster. The numbers on the arrows represent how many of the segments from the cluster on the left hand side were found in the cluster on the right hand side of the arrow.

The cluster solution with five clusters was used as the final solution, because the composition of the clusters changed only slightly between the five cluster and six cluster solution (see Figure 2); only a small new cluster was formed, while the other clusters remained more or less the same as in the five cluster solution. Also, in the five cluster solution, all clusters could be interpreted as a strategy. Tables 5 through 9 show, for each of the five clusters, the average transition probabilities that had the highest z-value, or a strikingly low or high absolute value, as well as the values for the performance variables and information about the number of actions. The interpretations of the clusters as strategies are provided below.

3.2.1 Cluster 1 – Little Help. The transition probabilities were generally low in this cluster, ranging from 0.00 to 0.08 (only the most salient transition probabilities are shown in Table 5), which means little help was requested. Only in 13% of the segments a student asksed for a solution, but the SolutionToStop variable had a mean and variance of zero, which means that even if the solution was given, the students continued to work on the exercise. Therefore, the solution was not used as an easy method of getting the answer. Also when the next step was requested, this step was almost never literally copied. The length of the exercises was rather short, with mean action count of 9.86. In 18% of the segments the users were successful in completing an exercise without consulting the answer. Fifty-two percent of the segments were in this cluster.

Table 5

Most Salient Transition Probabilities and Performance Variables for Cluster 1 (Little Help)

Variable	*M / Percentage*	*SD*	Z-value
	Transition Probabilities		
StepToCopystep	0.00	0.02	-0.40
StephintToStep	0.01	0.06	-0.40
SolutionToStop	0.00	0.00	-0.57
	Performance Variables		
ActionCount	9.86	7.46	-0.36
AskedSolution	13%		
Success	18%		

Note. N=141 (52%)

3.2.2 Cluster 2 – Click Through Help. As Table 6 shows, there are striking transition probabilities in StephintToStep (0.91), StrategyhintToStephint (0.24), and StepToSolution (0.19). These transition probabilities had the highest z-values, ranging from 0.83 to 2.80. SolutionToStop (0.36) had a high absolute value. These values suggest that the strategy was to start with the most general form of help, often continuing with asking for more specific help, sometimes even asking for the solution. In 83% of the segments the solution was asked for, only in 8% was the student successful. The number of actions per segment was on average 26.71. Nine percent of the segments were assigned to this cluster.

Table 6

Most Salient Transition Probabilities and Performance Variables for Cluster 2 (Click Through Help)

Variable	*M / Percentage*	*SD*	Z-value
	Transition probabilities		
StephintToStep	0.91	0.18	2.80
StrategyhintToStephint	0.24	0.35	0.89
StepToSolution	0.19	0.26	0.83
SolutionToStop	0.36	0.44	0.30
	Performance Variables		
ActionCount	26.71	25.23	0.44
AskedSolution	83%		
Success	8%		

Note. N=24 (9%)

3.2.3 Cluster 3 – Direct Solution. Not many segments were assigned to this cluster (8%), but the interpretation for this cluster is clear. Almost all transition probabilities were zero, the length of the segments was very small, and the high values for OkToSolution (0.91) and SolutionToStop (0.80) indicate that the strategy here was to start an exercise and immediately ask for the solution, after which the exercise was finished.

Table 7

Most Salient Transition Probabilities and Performance Variables for Cluster 3 (Direct Solution)

Variable	*M / Percentage*	*SD*	Z-value
	Transition probabilities		
OkToSolution	0.91	0.00	16.25
SolutionToStop	0.80	0.38	1.35
	Performance Variables		
ActionCount	4.14	1.93	-0.63
AskedSolution	100%		
Success	0%		

Note. N=21 (8%)

3.2.4 Cluster 4 – Step By Step. This is the only cluster in which the transition probability StepToCopystep was high (0.40). Other transition probabilities that were relatively or absolutely high were OkToStephint (0.38), OkToStep (0.25), NotokToStep (0.27), and StephintToStep (0.20), which indicates that the strategy was to use the step button quite often to find out the next step, and then to copy the code from that step. In 35% of the segments, the solution was asked for, but the low transition probability SolutionToStop (0.02) indicated that the given solution was not used to get the final answer. This cluster has the highest success rate; 35% of the exercises were successful. With an average of 46.07, the number of actions per segment was high in this cluster.

Table 8

Most Salient Transition Probabilities and Performance Variables for Cluster 4 (Step By Step)

Variable	*M / Percentage*	*SD*	Z-value
	Transition probabilities		
StepTocopyStep	0.40	0.27	1.61
OkToStephint	0.38	0.28	1.37
OkToStep	0.25	0.24	1.15
NotokToStep	0.27	0.35	1.04
StephintToStep	0.20	0.22	0.29
	Performance Variables		
ActionCount	46.07	30.78	1.35
AskedSolution	35%		
Success	35%		

Note. N=46 (17%)

3.2.5 Cluster 5 – Quick Solution. In this cluster, the transition probabilities StrategyhintToSolution (0.17), StephintToSolution (0.09), NotokToSolution (0.16), StepToSolution (0.16) had the highest z-scores (0.65 – 0.77). Furthermore, SolutionToStop

(0.97) was very high. These values indicate that often the solution was consulted, and that the students quit the exercise after getting the solution. The difference with cluster 3 (Direct Solution) is that other help functionalities were consulted too, before asking for the solution. In all cases the solution was eventually consulted, so there were no segments ending in success. We conclude that the strategy was to try the exercise for a while, ask for the solution, and then quit the exercise. The average number of actions per segment was 12.8, which is not very high.

Table 9

Most Salient Transition Probabilities and Performance Variables for Cluster 5 (Quick Solution)

Variable	*M / Percentage*	*SD*	Z-value
	Transition probabilities		
SolutionToStop	0.97	0.11	1.77
StrategyhintToSolution	0.17	0.37	0.77
StephintToSolution	0.09	0.25	0.69
NotokToSolution	0.16	0.31	0.68
StepToSolution	0.16	0.30	0.65
	Performance Variables		
ActionCount	12.18	9.26	-0.25
AskedSolution	100%		
Success	0%		

Note. N=39 (14%)

3.3 The Use of Strategies in Relation to Students and Exercises

The results of Fishers exact tests showed that the assignment of segments to clusters was related to the student who performed the segment, $p < .001$, and the exercise that was performed, $p < .01$, as expected. This means that students used a limited number of strategies, and that the type of exercise influenced the choice of strategy to some extent. Students who tried more than one exercise (n=48), sometimes only used a single strategy (23%) for all exercises, but more often two or three different strategies (23% and 39%), and sometimes even 4 strategies (14%). Students who used more than one strategy did not use all strategies equally often. To illustrate: 81% of students the students who tried more than one strategy used a single strategy for at least half of the exercises, while the other strategies were used much less often. This implies that even though students sometimes tried two or more different strategies, most of them seem to have a preference for one particular strategy.

3.4 Predicted Strategy Use by Achievement Goals

Table 10 shows the results of the backward logistic regression analyses. The use of the first strategy could not be predicted by any of the achievement goal scores. For the second strategy, R^2 = .20 (Cox & Snell), .28 (Nagelkerke), model χ^2 (1) = 10.32, p < 0.01. For the third strategy, R^2 = .19 (Cox & Snell), .30 (Nagelkerke), model χ^2 (1) = 9.83, p < 0.01. For the fourth strategy, the achievement goals were not significant predictors. For the fifth strategy, R^2 = .14 (Cox & Snell), .18 (Nagelkerke), model χ^2 (1) = 6.69, p = .15. The mastery approach contributes significantly to predicting the use of strategy 5. However, it could not be concluded that the model with predictors is an improvement compared to the model with no predictors.

Table 10

Results for the models for predicting strategies

					95% CI for Odds Ratio	
	B (SE)	*Wald*	*p*	*Odds Ratio*	*Lower*	*Upper*
	Use of Strategy 1 – Little Help					
Constant	1.16 (0.35)					
	Use Strategy 2 – Click Through Help					
Constant	-0.96 (0.40)					
Mastery Avoidance	1.55 (0.57)	7.53	.006	4.71	1.56	14.26
	Use Strategy 3 – Direct Solution					

Constant	-1.69 (0.50)					
Mastery Avoidance	-1.96 (0.84)	5.42	.02	0.14	0.03	0.73
Performance Avoidance	1.85 (0.78)	.561	.02	6.33	1.38	29.12
			Use Strategy 4 – Step By Step			
Constant	-1.04 (0.34)					
			Use Strategy 5 – Quick Solution			
Constant	0.14 (0.32)					
Mastery Approach	-4.89 (2.23)	4.82	.03	0.01	0.00	0.59
Mastery Avoidance	1.67 (1.00)	2.77	.10	5.30	0.74	37.80
Performance Avoidance	-2.13 (1.30)	2.68	.10	0.12	0.01	1.52
Performance Approach	1.94 (1.21)	2.56	.11	6.98	0.65	75.45

Note. CI = Confidence Interval. Only the final model of the backward method is shown. N = 45

4. Discussion and Conclusions

4.1 Help-Seeking Strategies and Achievement Goals

The aim of this study was to identify help-seeking strategies in an intelligent tutoring system, and to examine whether or not the use of these strategies could be predicted by achievement scores. The analyses resulted in five different strategies, three of which could be predicted by achievement goal scores.

In the first strategy, students seemed to attempt to finish the exercise with as little help as possible, because the number of help requests was low and the solution was in most cases not consulted. We call this strategy ‘Little Help’. In the second strategy students tended to start with some general form of help, but, presumably because the help was not sufficient, a more specific form of help was then requested, until it was clear for the student how to proceed. We call this strategy ‘Click Through Help’. In the third strategy, called ‘Direct Solution’, students started an exercise, clicked on the solution, and quit the exercise. In the fourth strategy, called ‘Step by Step’, we concluded that the students abused the help system by requesting each step and literally copying the code from each step until the exercise was finished. Finally, the last strategy, called ‘Quick Solution’, was somewhat similar to the

Direct Solution strategy, except that students tried the exercise first before they consulted the solution and quit.

For each strategy, a measure of success was calculated, but this should not be interpreted as a measure of learning effectiveness. For example, the highest success rate is found for the Step by Step strategy, but students who use this strategy requested every step of the solution. Also, the 0% success rates for the Direct Solution and Quick Solution strategies follow directly from the definition of the success measure and the strategy used, and do not directly imply that students did not learn anything from these strategies.

The findings seem to fit well with the current theories and evidence on help seeking. Over half of the strategies were identified as Little Help, which could indicate that the help functionality was underused. This is consistent with other findings about usage of help functionality in ILEs (Gräsel, Fischer, & Mandl, 2001; Aleven & Koedinger, 2000, Aleven et al. 2006, Köck & Paramythis, 2011). Interpreting the results in this way, however, might be premature. Stahl and Bromme (2009) found that a students can successfully adapt her help-seeking behavior to the complexity of the task. Therefore, if the exercises are easy, the amount of help requested might be appropriate. However, the low success rates (18%) with this strategy suggest otherwise; it seems that students could have benefitted from more help.

Why, then, do students not request more help? One explanation might be that students do not realize they need help. Becoming aware of the need of help is the first step in the feedback seeking process (Ryan et al, 2001), and students might lack the metacognitive skills to do so. Another possible explanation can be found in the concepts of threat to self-esteem and threat to autonomy. Some students might associate seeking help with a threat to self-esteem, even in a non-social situation or situations without penalties for help-seeking behavior (Karabenick, 2004), or seeking help conflicts with their need for autonomy (Huet, Escribe, Dupeyrat, Sakdavong, 2011). Another explanation for the lack of help seeking might be that students did not perceive the feedback as valuable. The perceived cost and value of feedback seeking are important determinants for the decision to seek help (VandeWalle & Cummings, 1997). This is in line with the Technology Acceptance Model (Davis, 1989), which shows that the perceived usefulness of technology influences the use of that technology.

Köck and Paramythis (2011) also found the Click Through Help strategy. Aleven et al. (2006) also find the Clicking Through Hints strategy. Baker, Walonoski, Heffernan, Roll, Corbett, & Koedinger (2008) find a strategy called ‘Gaming the System’, defined as “an attempt to succeed in an educational environment by exploiting properties of the system’s help and feedback rather than by attempting to learn the material”, which is similar to the Step by Step strategy. Furthermore, the Direct Solution and Quick Solution strategies are not very surprising, because if students have the option of consulting a solution, some of them will use it as a means to finish the exercise.

The difference between the Direct Solution and Quick Solution strategies might be interesting, and has not been reported before. We interpret this difference as follows: students who used the Direct Solution may have planned to ask for a solution before starting the exercise, making this purely executive help-seeking behavior. In contrast, students who used the Quick Solution strategy may have started the exercise *without* the intention to ask for a solution. Then, if they request the solution because they find the exercise too difficult, we find this appropriate help-seeking behavior. Consequently, this strategy can be classified as instrumental help seeking instead of executive. This means that although the two strategies look very much alike, they might represent two very different types of help-seeking behavior.

The above interpretation is supported by the finding that the use of these two strategies cannot be predicted by the same achievement goals. The Direct Solution strategy is negatively predicted by mastery avoidance and positively predicted by performance avoidance. We interpret this as follows. A student with a high learning avoidance goal, who typically wants to avoid learning less than is possible, will not directly ask for a solution because she feels this would prevent her from learning how to develop functional programs independently. A student with a high performance avoidance goals directly asks for the solution, because that allows her to see the solution quickly and makes sure she is not lagging behind the rest. The results indicated that the mastery approach negatively predicts the use of the Quick Solution strategy. This might mean that a student with a mastery approach goal refrains from asking for the solution too soon, because she feels she might learn more if she keeps trying instead.

The use of the Click Through Help strategy is positively predicted by mastery avoidance goals. We interpret this result as follows. A student with a high mastery avoidance goal starts with a general request for help because she aims to learn, but she often requests more specific help directly afterwards because she wants to avoid any misunderstandings.

The Little Help strategy could not be predicted by achievement goals. There are several possible explanations for this. First, many students might not have perceived the help functionality as useful, and this perception is not related to achievement goals. However Hofer et al. (1996) finds that even if a student reports that she finds a help system useful, she might still rarely use it. Another possible explanation is that a student with mastery goals as well as performance goals might have reasons not to seek help, because threat to self-esteem is related to performance goals, while threat to autonomy is related to mastery goals (Huet, Escribe, Dupeyrat, & Sakdavong, 2011). Finally, achievement goals did not predict the use of the Step By Step strategy.

The strategies and their relation to achievement goals are partially consistent with findings from Huet, Escribe, Dupeyrat, and Sakdavong (2011), who found that a student with high performance goals tends to seek less help. For our strategies this would mean that a student with high performance goals would use the Little Help strategy more often. This, however, was not the case. Huet et al. (2011) also found that a student with mastery goals perceives help seeking to be a threat to autonomy. This is in line with the reasoning explaining why a student with high mastery approach goals uses the Quick Solution strategy less often: she might feel that asking for a solution might prevent her from learning how to develop functional programs autonomously.

How do students actually learn when using the Intelligent Tutoring System? Student learning could take place through mechanisms of *self-explanation* when using an exercise in this tutor as a worked example or as a solved example problem (Chi et. al., 1989; Schworm & Renkl, 2006; Pirolli & Recker, 1994). According to Renkl and Atkinson (2003), a student could learn in an ILE when she explains steps in a solution by relating it to domain principles, thus generating a principle based understanding of a solution. A student could also focus on (sub)goals of a problem, and learn about the operators that are associated with those goals. Furthermore, a student could notice coherence between different examples to learn the abstraction of principles and schemas, which allows the student to solve novel problem too. Renkl (1997) also found that a student could learn effectively by applying anticipative reasoning, a strategy in which a student tries to anticipate the next step of the example before looking it up. It is worth studying whether the use of self-explanations is more plausible when students choose particular help-seeking strategies. An advantage of the tutor used in this research is that a student can decide herself to use an exercise as a worked example, a solved example, or as a problem solving exercise. Perhaps the use of self-explanations is more plausible when students choose a help-seeking strategy that could be considered as *a worked-out example* (Click Through Help and Step By Step strategies), which allows for explaining and anticipating solution steps. Furthermore, achievement goals may influence the use of self-explanation activities. A student with mastery goals, for instance, may be more willing to put in the necessary mental effort high quality self-explanations than a student with performance goals. Whether this is the case should be addressed in future research.

Finally, *cognitive load theory* (e.g. Merriënboer & Sweller, 2005) may partially explain students' help-seeking behavior in the tutor. Renkl and Atkinson (2003) described how the difficulty of the exercise and the prior knowledge of a student determine the *intrinsic* cognitive load in worked-out examples. Self-explanation and other learning activities are considered *germane* load, which supports learning. All other mental activities are *extraneous* load and do not directly support learning. A student may try to reduce extraneous load by choosing strategies like Click Through Help or Step By Step. However, if a student is in a more advanced skill acquisition phase, self-explanation may become unnecessary extraneous load, and a student may choose another help-seeking strategy such as the Little Help strategy. Also the role of cognitive load during the choice of help-seeking behavior may be addressed in further research.

4.2 Implications for Research and ITS Design

The approach we have taken in this paper is quite different from the approach taken by Aleven et al. (2006), who designed a normative model that decides whether or not help requests are desirable. This gives a measure of the quality of a student's help-seeking behavior, capable of predicting performance, something that we cannot do. A drawback of their method is that the model requires information about the knowledge level of the student on the subject studied, and it needs estimates of how much time it should take a student to complete a step, and whether or not she can finish the exercise without help. For a complex task such as designing a functional program, it is hard to get good estimates. Our approach can more easily be applied to a variety of ITSs in future research, as it does not depend on these estimates.

A number of our findings are relevant for designers of ITSs, and for teachers using ITSs. First, many students probably request less help than they need. Anticipating on this behavior, designers might add help-seeking hints to their tutor (c.f. Roll, Aleven, McLaren, & Koedinger, 2010), and improve the accessibility and usability of the help functionality.

Teachers might encourage students to use the help functionality when necessary. Second, we want to avoid students consulting a too quickly, as this might hamper learning. Designers might consider not offering this option in a tutor, or restricting access to it. For the designers of the functional programming tutor (Jeuring, Gerdes, & Heeren, 2011), the results indicate that the strategy hint and the step hint may be perceived as unhelpful, because there are no strategies in which students successfully finish an exercise using these forms of help.

4.3 Limitations

There are some limitations to the current research. Even though the initial sample size was sufficient, complete data on the use of strategies and achievement goals was available for 45 participants, due to attrition. With a regression analysis with 4 predictors and a sample size of 45, the power for large effect sizes is sufficient (.86), but for medium effect sizes it is low (.47) (Faul, Erdfelder, Buchner, & Lang, 2009). This might partially explain why achievement goals do not predict the use of some strategies.

We measured help-seeking behavior over the span of a single exercise. However, some students may have used more than one strategy within an exercise, which may have resulted in less reliable results. Furthermore, the DMMs we used to model user behavior are limited to sequences of two consecutive actions. In future research, this might be increased to longer sequences. There are several other methods for modeling behavior that are more informative in some cases (Romero & Venture, 2006). However, these methods are more complex, and DMMs offer a good balance between simplicity and depth of information (Soller and Lesgold, 2007).

The selection of transition probabilities might be considered subjective. However, we carefully selected the variables based on theory on help seeking and on their interpretability in terms of help seeking. We prefer to use theoretically justified variables over computationally selecting ‘meaningful’ variables (Steinley, 2006; Köck & Paramythis, 2011).

Finally, the *k*-means clustering method used to identify strategies has some inherent weaknesses. First, the algorithm terminates at a local optimum, which is never verifiable globally optimal. Therefore, we ran the algorithm 250 times for each solution, and selected the best result. Another issue is that we did not know the number of clusters in advance. We could not determine this number automatically, nor can we prove that there *is* an exact number of strategies that is used. We inspected multiple solutions, and found that most clusters from a particular solution were similar to clusters in the solution with one cluster more or less. Hence a slightly different choice in the number of clusters would not produce completely different results.

Despite the above limitations, we believe the results are valid for the following reasons. First, all cluster centers could be interpreted and linked to theory or previous research, and some of the clusters were related to achievement goals, as expected. The use of strategies was related to a student as well as an exercise, which we expected, and the clusters clearly differentiated in terms of performance and number of actions per exercise.

4.4 Future Research

Our approach should be applied to a variety of ITSs to contribute to theoretical models on help seeking. It would be interesting to see whether or not there exists an overlap between strategies in different ITSs and their relation to achievement goals, even if the ITSs differ in their design and the context in which they are used. Such results would contribute to a more general and accurate model of help-seeking behavior in ITSs. Furthermore, it would be interesting to see if other personal factors, such as self-efficacy or prior knowledge, can be linked to help-seeking strategies (Bartholomé, Stahl, Pieschl, Bromme, 2005). Finally, it should be investigated whether or not the use of our strategies can be related to learning mechanisms and learning gains, and if it possible to improve students' help-seeking strategies, for example by automated feedback (see also Aleven, Roll, and Koedinger, 2010).

5. Conclusions

We have presented the first empirically based, fully documented list of help-seeking strategies that students used in an ITS, as well as the relation between the use of these strategies and achievement goals. Designers of ITSs can use our findings to offer functionality that targets only certain types of users and encourages constructive use of the help functions. Our results can also be used as a first step towards devising a more general, empirically based model for help-seeking behaviors in ITSs, which may contribute to theoretical knowledge on help seeking. Furthermore, because we determined specific strategies and the classification is done automatically, our research supports researchers in studying help-seeking behavior and its relation to other variables.

References

Aleven, V., Mclaren, B. M., Roll, I., & Koedinger, K. (2006). Toward Meta-cognitive Tutoring: A Model of Help Seeking with a Cognitive Tutor. *International Journal of Artificial Intelligence, 16*(2), 101-128.

Aleven, V., Roll, I. D. O., McLaren, B. M., & Koedinger, K. R. (2010). Automated, Unobtrusive, Action-by-Action Assessment of Self-Regulation During Learning With an Intelligent Tutoring System. *Educational Psychologist, 45*(4), 224-233. doi: 10.1080/00461520.2010.517740

Aleven, V., Stahl, E., Schworm, S., Fischer, F., & Wallace, R. (2003). Help Seeking and Help Design in Interactive Learning Environments. *Review of Educational Research, 73*(3), 277-320. doi: 10.3102/00346543073003277

Ames, R., & Lau, S. (1982). An attributional analysis of student help-seeking in academic settings. *Journal of Educational Psychology, 74*(3), 414-423. doi: 10.1037/0022-0663.74.3.414

Arbreton, A. (1998). Student Goal Orientation and Help-Seeking Strategy Use. In A. A. Karabenick (Ed.), *Strategic Help Seeking. Implications for Learning and Teaching* (pp. 95 - 116). London: Lawrence Erlbaum Associates.

Atkinson, R. K., Derry, S. J., Renkl, A., & Wortham, D. (2000). Learning from Examples: Instructional Principles from the Worked Examples Research. *Review of Educational Research, 70*(2), 181-214. doi: 10.3102/00346543070002181

Baker, R., Walonoski, J., Heffernan, N., Roll, I., Corbett, A., & Koedinger, K. (2008). Why students engage in "gaming the system" behavior in interactive learning environments. *Journal of Interactive Learning Research, 19*(2), 185-224.

Bartholomé, T., Stahl, E., Pieschl, S., & Bromme, R. (2006). What matters in help-seeking? A study of help effectiveness and learner-related factors. *Computers in Human Behavior, 22*(1), 113-129. doi: 10.1016/j.chb.2005.01.007

Beal, C. R., Qu, L., & Lee, H. (2008). Mathematics motivation and achievement as predictors of high school students' guessing and help-seeking with instructional software. *Journal of Computer Assisted Learning, 24*(6), 507-514. doi: 10.1111/j.1365-2729.2008.00288.x

Bong, M. (2009). Age-related differences in achievement goal differentiation. *Journal of Educational Psychology, 101*(4), 879-896. doi: 10.1037/a0015945

Chi, M. T. H., Bassok, M., Lewis, M. W., Reimann, P., & Glaser, R. (1989). Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science, 13*(2), 145-182. doi: http://dx.doi.org/10.1016/0364-0213(89)90002-5

Davis, F. D. (1989). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. (T. Ball & R. B. Jones, Eds.) *MIS Quarterly*, 13(3), 319. JSTOR. Retrieved from http://www.jstor.org/stable/249008?origin=crossref

Deci, E. L., & Ryan, R. M. (1987). The support of autonomy and the control of behavior. *Journal of Personality and Social Psychology, 53*(6), 1024-1037. doi: 10.1037/0022-3514.53.6.1024

Elliot, A. J., & McGregor, H. A. (2001). A 2 × 2 Achievement Goal Framework. *Journal of Personality and Social Psychology, 80*(3), 501-519.

Elliot, A. J., & Murayama, K. (2008). On the measurement of achievement goals: Critique, illustration, and application. *Journal of Educational Psychology, 100*(3), 613-628. doi: 10.1037/0022-0663.100.3.613

Faul, F., Erdfelder, E., Buchner, A., & Lang, A.-G. (2009). Statistical power analyses using G*Power 3.1: Tests for correlation and regression analyses. *Behavior Research Methods, 41*, 1149-1160.

Gall, S. N.-L. (1981). Help-seeking: An understudied problem-solving skill in children. *Developmental Review, 1*(3), 224-246. doi: 10.1016/0273-2297(81)90019-8

Gräsel, C., Fischer, F., & Mandl, H. (2001). The use of additional information in problem-oriented learning environments. *Learning Environments Research*, 3, 287–305.

Graesser, A. C., Chipman, P., Haynes, B. C., & Olney, A. (2005). AutoTutor: an intelligent tutoring system with mixed-initiative dialogue. *Education, IEEE Transactions on, 48*(4), 612-618. doi: 10.1109/te.2005.856149

Hofer, M., Niegemann, H. M., Eckert, A., & Rinn, U. (1996). Pädagogische Hilfen für interaktive selbstgesteuerte Lernprozesse und Konstruktion eines neuen Verfahrens zur Wissensdiagnose [Instructional help for interactive self-directed learning processes and construction of a new procedure for knowledge diagnosis]. *Zeitschrift für Berufs- und Wirtschaftspädagogik Beiheft*, 13, 53–67.

Huet, N., Escribe, C., Dupeyrat, C., & Sakdavong, J.-C. (2011). The influence of achievement goals and perceptions of online help on its actual use in an interactive learning environment. *Computers in Human Behavior, 27*(1), 413-420. doi: 10.1016/j.chb.2010.09.003

Jeuring, J., Gerdes, A., & Heeren, B. (2011). A programming tutor for Haskell. *Proceedings of CEFP 2011: Lecture Notes of the Central European School on Functional Programming:* LNCS 7241, 1-45, Springer. doi: 10.1007/978-3-642-32096-5_1

Karabenick, S. A. (2004). Perceived Achievement Goal Structure and College Student Help Seeking. *Journal of Educational Psychology, 96*(3), 569-581. doi: 10.1037/0022-0663.96.3.569

Karabenick, S. A. (2011). Classroom and technology-supported help seeking: The need for converging research paradigms. *Learning and Instruction, 21*(2), 290-296. doi: 10.1016/j.learninstruc.2010.07.007

Köck, M., & Paramythis, A. (2011). Activity sequence modelling and dynamic clustering for personalized e-learning. *User Modeling and User-Adapted Interaction, 21*(1), 51-97. doi: 10.1007/s11257-010-9087-z

Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education, 8*, 30-43.

Kuhl, J. (1985). Volitional mediators of cognitive-behavior consistency: Self-regulatory processes and action versus state orientation. In J. Kuhl & J. Beckman (Eds.), *Action control* (pp. 101 - 128). New York: Springer-Verlag.

Maehr, M. L. (1989). Thoughts about motivation. In C. Ames & R. Ames (Eds.), *Research on Motivation in Education* (Vol. 3). New York: Academic Press.

Merriënboer, J. G., & Sweller, J. (2005). Cognitive Load Theory and Complex Learning: Recent Developments and Future Directions. *Educational Psychology Review, 17*(2), 147-177. doi: 10.1007/s10648-005-3951-0

Renkl, A. (1997). Learning from worked-out examples: A study on individual differences. *Cognitive Science, 21*(1), 1-29. doi: http://dx.doi.org/10.1016/S0364-0213(99)80017-2

Pirolli, P., & Recker, M. (1994). Learning Strategies and Transfer in the Domain of Programming. *Cognition and Instruction, 12*(3), 235-275. doi: 10.1207/s1532690xci1203_2

Renkl, A., & Atkinson, R. K. (2003). Structuring the Transition From Example Study to Problem Solving in Cognitive Skill Acquisition: A Cognitive Load Perspective. *Educational Psychologist, 38*(1), 15-22. doi: 10.1207/S15326985EP3801_3

Roll, I., Aleven, V., McLaren, B. M., & Koedinger, K. R. (2011). Improving students' help-seeking skills using metacognitive feedback in an intelligent tutoring system. *Learning and Instruction, 21*(2), 267-280. doi: 10.1016/j.learninstruc.2010.07.004

Romero, C., Ventura, S., Espejo, P.G., Hervás, C.: Data mining algorithms to classify students. In: *Proceedings of the 1st International Conference on Educational Data Mining* (EDM08), pp. 8–17, (2008)

Ryan, A., Pintrich, P., & Midgley, C. (2001). Avoiding Seeking Help in the Classroom: Who and Why? *Educational Psychology Review, 13*(2), 93-114. doi: 10.1023/a:1009013420053

Soller, A., Lesgold, A.:Modeling the process of collaborative learning. In:Dillenbourg, P.,Hoppe, H., Ogata, H., Soller, A. (eds.) *The Role of Technology in CSCL*, vol. 9 of Computer-Supported Collaborative Learning, pp. 63–86. Springer, New York (2007)

Stahl, E., & Bromme, R. (2009). Not everybody needs help to seek help: Surprising effects of metacognitive instructions to foster help-seeking in an online-learning environment. *Computers & Education, 53*(4), 1020-1028. doi: 10.1016/j.compedu.2008.10.004

Steinley, D. (2006). K-means clustering: A half-century synthesis. *British Journal of Mathematical and Statistical Psychology, 59*(1), 1-34. doi: 10.1348/000711005x48266

Schworm, S., & Renkl, A. (2006). Computer-supported example-based learning: When instructional explanations reduce self-explanations. *Computers & Education, 46*(4), 426-445. doi: http://dx.doi.org/10.1016/j.compedu.2004.08.011

Tibshirani, R., Walther, G., & Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 63*(2), 411-423. doi: 10.1111/1467-9868.00293

VandeWalle, D., & Cummings, L. L. (1997). A test of the influence of goal orientation on the feedback-seeking process. *Journal of Applied Psychology, 82*(3), 390-400. doi: 10.1037/0021-9010.82.3.390

Vanlehn, K., Lynch, C., Schulze, K., Shapiro, J. A., Shelby, R., Taylor, L., . . . Wintersgill, M. (2005). The Andes Physics Tutoring System: Lessons Learned. *Int. J. Artif. Intell. Ed., 15*(3), 147-204.

Wood, H., & Wood, D. (1999). Help seeking, learning and contingent tutoring. *Computers and Education, 33*(2), 153-169. doi: 10.1016/s0360-1315(99)00030-5