Global Feedback in ACTIVEMATH

Erica Melis and Eric Andrès
DFKI and Universität des Saarlandes
Universität des Saarlandes
Stuhlsatzenhausweg nn
66123 Saarbrücken, Germany melis@activemath.org

Abstract. This paper describes the global feedback in ACTIVEMATH, a learning environment for mathematics and beyond. We describe the blackboard-architecture with its components and easily modifiable rules and present the user-adaptive global learning suggestions implemented so far.

1 Introduction

ACTIVEMATH is a user-adaptive, web-based learning environment for mathematics. It generates learning material for the individual learner according to her learning goals, preferences, and mastery of concepts as well as to the chosen learning scenario [7].

ACTIVEMATH's user model consists of the history data base, the user's preferences profile, and a database of mastery-levels of the concepts in the domain. The history contains information about the user's activities (reading time for instructional items, exercise success rate, manual changes of the user model). The user profile contains the student's preferences, learning scenario, and learning goals submitted for a session. To represent the concept mastery assessment, currently the user model contains values for a subset of the competences of Bloom's taxonomy [1]: knowledge, comprehension, and application. Bloom defines the skills needed for 'knowing' as recall of information, knowledge of major ideas; we assume that knowledge-mastery is reached by reading but also from working with examples and exercises. Bloom defines comprehension skills as understanding information, translating knowledge into a new context, predicting consequences; in ACTIVEMATH we assume that comprehension-mastery can be achieved by understanding examples and relating several concepts and then answer questions. Bloom defines the needed skills as usage of the concept (in new situations), solving problems using the required skills or concepts; we diagnose application-mastery from performance in exercises in which a concept is applied. Similar categorizations of mastery have been employed by several researchers, e.g., Shute [10] in her taxonomy of outcome types.

Finishing an exercise or navigating to another page triggers an updating of the user model in ACTIVEMATH. A diagnosis of different types of user actions updates the values of the different competencies.

So far, ACTIVEMATH primarily served the adaptive generation of documents, the integration of service systems, and local feedback in interactive exercises [2]. This paper focuses on a new global suggestion mechanism.

The plan of the paper is as follows. After some preliiminaries we distinguish different types of feedback, local and global feedback. Then, the global feedback is described and

substantiated, in particular, the architecture and some of the user-adaptive global learning suggestions implemented so far.

Notation. Note that K/C/A-present(c) is an abbreviation for: presentation of content contributing to the concept c in a K-/, C-/, or A-oriented way respectively, where K-oriented means present just concepts and possibly explanations; C-oriented means present concepts and examples; A-oriented means present the full spectrum of content including concepts, examples, and exercises. K/C/A-present(c) are functions of the ACTIVEMATH' existing course generator. Moreover, the X in the diagnostic expression `seenButUnknown`(X) denotes the level of mastery of the focus-concept that is missing. The focus-concept is the concept that is elaborated by the material currently presented to the student.

1.1 Local and Global Feedback

Usually, feedback and help in intelligent tutoring systems (ITSs) is designed for a direct response to students' problem solving actions and it is designed in order to help students to accomplish a solution. This *local* feedback can comprise [1]

- 'Knowledge of result', i.e., a feedback that states whether a solution is correct or not. For good students it might suffice to stimulate further elaboration.
- 'Knowledge of correct result' provides a correct solution. This is recommended for students with little prior knowledge, little ability, many errors, and relatively simple learning goals.
- 'Answer until correct' asks the student to try again until she gives a correct answer. It proved valuable, for complicated task and students with sufficient ability to solve the exercise.
- 'Instruction-based elaboration' may include explanation of the correct solution, correction of errors, etc.

We think that two kinds of feedback and guidance can be provided by an ITS, a *local* response to student activities which is supposed to coach the correction of a problem solving attempt of the learner and a *global* feedback coaching (several aspects of) the entire learning process.

In ACTIVEMATH, local and global feedback is distinguished because of their different aims, different learning dimensions, and different mechanisms. In addition, our usage of service systems for the check of the correctness of problem solving steps and for the generation of local problem solving feedback is a practical reason for separating local and global feedback. The local feedback such as 'syntax error', 'step not correct, because...', 'task not finished yet', or 'step not applicable' is computed with the help of a system and related to a problem solving step in an exercise or to the final achievement in an exercise. In what follows we deal with global feedback only.

[1] this summary is adopted from [4]

2 Global Feedback in ACTIVEMATH

Currently, the global feedback in ACTIVEMATH does not require specific authoring because it focuses on general suggestions on navigation through the hypertext material, repetition of little understood concepts or exercises.

The computation of global feedback requires diagnoses for several types of user activities. The information about the student's navigation, her reading, understanding, and problem solving actions, e.g. the duration and success rate, serves as a basis for the user-adaptive suggestions. That is, information from the history of the learner's actions and information about her mastery is necessary to generate useful suggestions.

2.1 Blackboard Architecture

The architecture for the global suggestion mechanism in ACTIVEMATH clearly separates diagnoses and suggestions as shown in in Figure 1. An advantage of this separation is that the same diagnosis results can be used by different suggestion mechanisms, in different pedagogical strategies, etc. For instance, if the diagnosis yields a `seen(example, insufficient)`,[2] then `example` could be presented again in a strict-guidance strategy but not in a weak-guidance strategy.

Figure 1 shows several evaluators (diagnosis agents) which pass their results to a *diagnosis blackboard* (DBB) and to the user model that updates the user's mastery-level of concepts and the activity history. *Intermediate* diagnoses are computed by other evaluators from the information on the DBB and in the user model. These diagnosis are written on the DBB too.

As displayed in Figure 1, several suggestors compute global feedback from the diagnoses on the DBB and pass their results to the *suggestion blackboard* (SBB). If necessary, the results are sent to a ConflictManager that rates the different suggestions on the SBB. Then, the best rated suggestions are executed.

Some evaluators provide a diagnosis immediately from one of a user's action while other (intermediate) evaluators infer a diagnosis from the DBB and additional information. In ACTIVEMATH, each of the immediate evaluators watch one of the following types of activities

- navigation
- reading (time)
- problem solving (assessed performance)
  - multiple choice questions (MCQ) exercises
  - exercises with a Computer Algebra System
  - exercises with the Omega proof planner.

New immediate or intermediate diagnosis agents can be easily added, e.g., an evaluator for the individual average reading time.

[2] i.e., the time for reading the example is less than a threshold

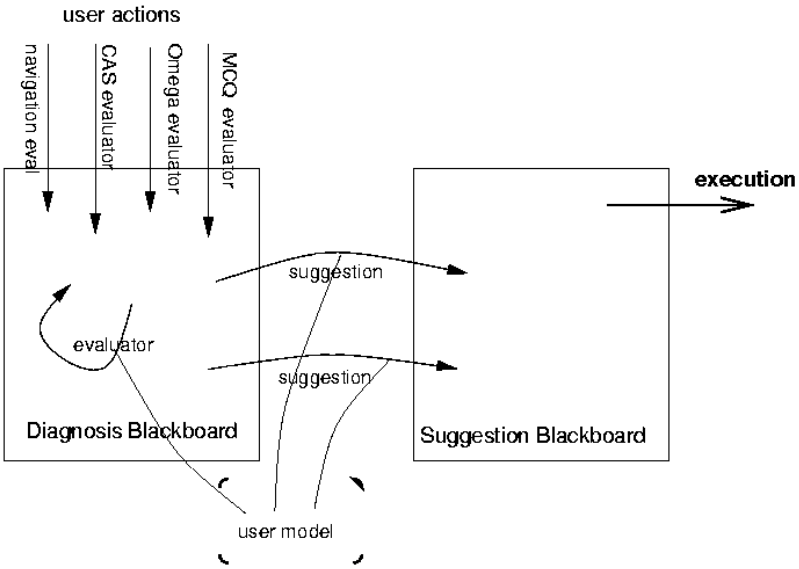

Figure 1: The architecture of evaluator and suggestion mechanisms

2.2 The Global Suggestion Mechanisms in ACTIVEMATH

Based on the facts on the SBB, the actual suggestion presentation is generated to deliver

- a neutral, happy (smile), or a sad (noSmile) face which is thought as a shortcut. The default expression is neutral.
- the verbal feedback (only abbreviated in the tables below) and
- one or several presentation-actions.

For each learning-goal level (that is, K-, C-, or A), a *suggestion strategy* can be designed as a set of suggestors. In what follows, we present suggestors of an *A-level* oriented suggestion strategy which (1) reacts to navigation problems and (2) suggests content.

The suggestors can be implemented as agents or as rules writing on a blackboard. The current implementation is with rules for the Java expert system shell Jess [3]. For more details see [6].

1. navigation help
2. content suggestions
    - present new or skipped example
    - present similar example
    - present new exercise
    - present same exercise
    - present again the focus-concept maybe also examples, exercises

- present (missed) instructional items[3]
- present certain prerequisites of the focus-concept maybe together with examples and exercises
- K-present a concept. This means a presentation of items that intends the improvement of the user's knowlege of a concept. Mainly definitions and elaborations will be shown.
- C-present a concept. The goal of this action is to lead the user to an understanding of the concept. Currently, this is tried by presenting items like examples and comprehension questions.
- A-present a concept, i.e., present some exercises and other items that help the user to learn how to apply the concept.

First, we explain the essence of a rule and then a presentation of the actual rules follows. The right-hand-side of the rule presentations contain a face-expression, an (abbreviated) verbal feedback, and the suggested presentation actions.

Rules for Navigation Suggestions These rules are needed because ACTIVEMATH delivers a hypertext learning document and it is known that navigation in hypertexts needs special attention [8] since being lost in hyperspace puts an additional load on the learner.

- if the user navigated appropriately, then provide reassuring feedback (abbreviated by 'smile').

```
IF Navig(okay) THEN - smile -
```

- if an irrational navigation is diagnosed that started at point `?start` of the table of contents, then two pointers show the current position and the `?start` position. In this case, the user can click the `?start` position to return to a 'useful' learning path.

```
IF Navig(irrational,?start) THEN - noSmile -
                                 ''did you get lost by chance?''
                                 pointer(current) and pointer(?start)
```

Rules for Content Suggestions are needed because if the goal-level of mastery is not yet reached by the learner, then the presentation of appropriate content might help to improve. As opposed to the local feedback that corrects single problem solving steps, the global feedback described below prompts and supports the learner in activities such as reading, repeating, self-explaining, comparing, varying, information gathering that are known to improve learning, see, e.g., [9, 12].

- Even if the mastery-level is reached, more exercising can strengthen the mastery of the concept and the confidence for less confident learners. Therefore, if there is an exercise that is more difficult than those solved already, then it can be offered.

[3] if misunderstanding of concept not attributed to prerequisites or too little reading time

```
IF Known(A ?focus)                              THEN - smile -
AND solution(?id correct)                            ‘‘see more’’
AND notConfident                                     present(?exc1)
AND exerciseFor(?focus ?id)
AND exerciseFor(?focus moreDiff(?id ?exc1))
AND notSeen(?exc1)
```

- If `seenButUnknown(A)` holds for the focus-concept, then present examples similar to the failed exercise of the focus-concept, unseen simpler exercises, and then again the incorrectly solved exercise of the focus-concept.

  This suggestion is made because A-mastery is the learning goal but not yet achieved, and therefore another exercise for the focus-concept should be offered to be solved. This exercise should be a bit simpler in order to keep the user’s motivation up (in the proximal zone of development). Then an example similar to the exercise should be shown for comparison. Finally, the originally failed exercises should be presented again.

```
IF SeenButUnknown(A ?focus)        - noSmile -
AND solution(?id incorrect) THEN   ‘‘go deeper into the concept’’
AND exerciseFor(?focus ?id)        exerciseFor(?focus lessDiff(?id ?exc1))
                                   exampleFor(?focus simTo(?id ?exm1))
                                   present(?exm1 ?exc1 ?id)
```

- If seenButUnknown(C) holds for the focus-concept, then show not yet sufficiently seen examples and ask for an explanation of the examples.

  This suggestion is made because C-mastery of the focus-concept is not yet achieved. To foster comprehension, an example that has not been studied (sufficiently) is presented and the learner is asked to engage herself in (self-)explanation.

```
IF SeenButUnknown(C ?focus)         - noSmile -
AND exampleFor (?focus ?exm) THEN   ‘‘please explain example’’
AND NOT Seen(okay ?exm)             present(?exm)
```

- If the focus-concept is known only insufficiently, then the user should re-read the last page on which the concept has been elaborated. Since A-mastery is the learning goal of the strategy, everything (reading, comprehending, and exercising) for the focus-concept, except any solved exercises, needs to be repeated. Certainly, this rule could be augmented with the help of more elaborate diagnoses that suggest the deeper reasons (e.g., distraction) why the student has seen enough but does not know anything.

```
IF seenButUnknown(K ?focus) THEN - noSmile -
                                 ‘‘please return to last page’’
                                  present(lastPage)

solution(?id correct)   THEN  REJECT present(?id)
```

- If notSeenAndUnknown holds for any of knowledge-, comprehension-, or application-level, then K/C/A-present, respectively. This suggestion is made because A-mastery is the learning goal and so every level not mastered yet is tackled.

```
IF notSeenAndUnknown(K/C/A) THEN - noSmile -
                                 ‘‘repeat’’
                                  K/C/A-present(focus)
```

- Siimilarly, missing prerequisites are handled at each level. If a prerequisite c is not yet mastered at one of the mastery-levels, then c will be presented according to what is missing for that mastery.

```
IF missingPre(?c, ?level)   THEN -noSmile -
                                 ‘‘missing prerequisite’’
                                 ?level-present(?c)
```

- If an item ID has been seen sufficiently and more than once, then do not present ID again without a particular need (rule). Otherwise the motivation might drop.

Conflict Management It is possible that conflicting suggestions occur on the SBB, in case a user action triggers diagnoses for which several rules fire. For instance, when a new page is selected by the user, this can trigger a navigation suggestion as well as a conflicting concept presentation suggestion. If not all of the suggestions can be presented at the same time, then a rating has to indicate the priorities, and only the rules with the highest rating will generate their suggestions.

The Suggestion User Interface We specified the reassuring 'smile' feedback as part of the user interface because we feel that reassurance and positive feedback is important for motivational reasons [5] and for avoiding situations in which the learner feels insecure. The primary user interface of the global suggestion mechanism consists of a face/companion with a variety of possible types of states. This face moves regularly even if the user does well and there is no new suggestion. This is implemented in order to prevent the user wondering about the mechanism being stuck. This face informs the user whether the system wants to offer a suggestion or not. If not, it reassures the learner by looking happily. When new coaching is suggested by the system, the face changes and the student can choose to follow or to ignore the suggestion. The offer expires after some time and disappears.

3 Conclusion

This article mainly describes global feedback that targets the overall learning. The description of ACTIVEMATH' global suggestion mechanism includes the architecture, some suggestion sources, and first results for the user interface. A more comprehensive account can be found in the technical report [6]. An on-line demo of ACTIVEMATH is available at `http://www.activemath.org`

Since diagnosis and suggestion agents can be easily exchanged and configured, different suggestion strategies can be devised and experimented with. We offer this tool to the research community. We only started the development and test of useful suggestions. Certainly, we shall investigate motivational diagnoses and suggestors [11] as well as more elaborate suggestion mechanisms and dialogues for presenting concepts, examples, and exercises. We shall also investigate a more personlized verbalization and its effects.

We thank the ACTIVEMATH group, in particular, Carsten Ullich and Sabine Hauser for involved discussions as well as Bernhard Jacobs for his supply of empirical knowledge.

References

[1] B.S. Bloom, editor. *Taxonomy of Educational Objectives: The Classification of Educational Goals: Handbook I, Cognitive Domain*. Longmans, Green, Totonto, 1956.

[2] J. Buedenbender, E. Andres, A. Frischauf, G. Goguadze, P. Libbrecht, E. Melis, and C. Ullrich. Using computer algebra systems as cognitive tools. In S.A. Cerri, G. Gouarderes, and F. Paraguacu, editors, *6th International Conference on Intelligent Tutor Systems (ITS-2002)*, number 2363 in Lecture Notes in Computer Science, pages 802–810. Springer-Verlag, 2002.

[3] E. Friedman-Hill. *Jess, the Java Expert System Shell*, 2000. `http:/herzberg.ca.sandia.gov/jess/`.

[4] B. Jacobs. Aufgaben stellen und Feedback geben. Technical report, Medienzentrum der Philosophischen Fakultät der Universität des Saarlandes, 2001.

[5] H. Mandl, H. Gruber, and A. Renkl. *Enzyklopädie der Psychologie*, volume 4, chapter Lernen und Lehren mit dem Computer, pages 436–467. Hogrefe, 1997.

[6] E. Melis and E. Andres. Evaluators and suggestion mechanisms for ACTIVEMATH. Technical report, DFKI, 2002.

[7] E. Melis, J. Buedenbender, E. Andres, A. Frischauf, G. Goguadse, P. Libbrecht, M. Pollet, and C. Ullrich. ACTIVEMATH: A generic and adaptive web-based learning environment. *Artificial Intelligence and Education*, 12(4):385–407, winter 2001.

[8] T. Murray, J. Piemonte, S. Khan, T. Shen, and C. Condit. Evaluating the need for intelligence in an adaptive hypermedia system. In G. Gauthier, Claude Frasson, and K. VanLehn, editors, *Intelligent Tutoring Systems, Proceedings of the 5th International Conference ITS-2000*, volume 1839 of *LNCS*, pages 373–382. Springer-Verlag, 2000.

[9] A. Renkl. Worked-out examples: Instructional explanations support learning by self-explanations. *Learning and Instruction*, 2001.

[10] V.J. Shute. SMART: Student modeling approach for responsive tutoring. *User Modeling and User-Adapted Interaction*, 5:1–44, 1995.

[11] T. Del Soldato. Detecting and reacting to the learner's motivational state. In , editor, *Proceedings of the International Conference on Intelligent Tutoring Systems*, volume 608 of *Lecture Notes in Computer Science*, pages 567–574. Springer, 1992.

[12] R. Stark. Instruktionale Effekte beim Lernen mit unvollständigen Lösungen. Forschungsberichte 117, Ludwigs-Maximilian-Universität, München, Lehrstuhl für Empirische Pädagogik und Pädagogische Psychologie, 2000.