Deriving Acquisition Principles from Tutoring Principles

Jihie Kim and Yolanda Gil

Information Sciences Institute, University of Southern California
4676 Admiralty Way, Marina del Rey, CA 90292, U.S.A.
{jihie, gil}@isi.edu

Abstract. This paper describes our analysis of the literature on tutorial dialogues and presents a compilation of useful principles that students and teachers typically follow in making tutoring interactions successful. The compilation is done in the context of making use of those principles in building knowledge acquisition interfaces since acquisition interfaces can be seen as students acquiring knowledge from the user. We plan to use these ideas in our future work to develop more proactive and effective acquisition interfaces.

1 Introduction

Transferring knowledge from humans to computers has proven to be an extremely challenging task. Over the last two decades, an array of approaches to interactive knowledge acquisition have been proposed. Some tools accept rules and check them against other existing rules [13, 22]. Some tools acquire knowledge suitable for specific tasks and problem solving strategies [29]. Other tools focus on detecting errors in the knowledge specified by the user [21, 23, 31]. Some systems use a variety of elicitation techniques to acquire descriptive knowledge [18, 35] often in semi-formal forms. There are some isolated reports of users with no formal background in computer science that are now able to use acquisition tools to build sizeable knowledge bases [24, 14, 10]. However, the majority of the burden of the acquisition task still remains with the user. Users have to decide what, how, and when to teach the system. Current acquisition tools do not take the kind of initiative and collaborative attitude that one would expect of a good student, mostly reacting to the user's actions instead of being proactive learners.

We set off to investigate how the dynamics of tutor-student interactions could be used to make acquisition tools better students to further support users in their role of tutors of computers. Given the success in deploying educational systems in schools and their reported effectiveness in raising student grades [26], we expected the tutoring literature to have useful principles that we could exploit. Another strength of tutoring work is that it is typically motivated by extensive analysis of human tutorial dialogues [17], which the knowledge acquisition literature lacks.

This paper describes our analysis of the literature on tutorial dialogues and presents a compilation of useful principles that students and teachers follow in making tutoring interactions successful and that could be useful in the context of interactive acquisition tools. We plan to use these ideas in our future work to develop more proactive acquisition interfaces.

The paper begins with a discussion of the similarities and differences between instructional systems (educational software and human tutoring) and interactive acquisition tools. We then present fourteen learning principles that we believe can be immediately incorporated into our current tools. Finally, we describe how acquisition interfaces can interact with users using these principles.

2 Tutorial Dialogues in Instructional Systems and in Interactive Knowledge Acquisition

In instructional systems (both educational software and intelligent tutoring systems), the tutor's role is to help the user (student) achieve some degree of proficiency in a certain topic (the lesson). In interactive acquisition interfaces, these roles are reversed. Acquisition tools can be seen as students learning new knowledge from the user (teacher) and they should be able to use some of the strategies that good learners pursue during a tutoring dialogue. Ideally, it should also be able to supplement the user's skills as a teacher by helping the user pursue effective tutoring techniques. This would help the user teach the material better and faster to the system, as well as delegate some of the tutor functions over to the system.

In essence, we are trying to investigate what it takes to create a good student, while most ITS work has focused on creating good teachers. We believe that the work in educational systems and acquisition systems share a lot of issues and they may be able to contribute to each other in many ways. In fact there has been work that bridges these two communities. For example, there have been recent interests in acquiring knowledge for intelligent tutoring systems [34]. We think that technology built by the knowledge acquisition community will be useful for building tools to help users develop the knowledge and models used in ITS.

There are some issues that interactive acquisition interfaces will not face. Human students in need of tutoring often have a lack of motivation that the instructional system has to address [28]. Instructional systems need to use special tactics to promote deep learning, such as giving incremental hints instead of showing the student the correct answers. Finally, our student will not be subject to the cognitive limitations of a typical human student, and can exploit memory and computational skills that would be exceptional (if not infrequent) for human students.

3 Principles in Teaching and Learning

We have been investigating various tutoring principles[1] used by human tutors and educational software [16, 41, 17]. Although human tutors provide more flexi-

[1] In the tutoring literature these are often referred to as tutoring strategies. We prefer to refer to them as tutoring principles, since we found that they can be implemented

ble support, the tutoring principles supported by educational software are often inspired by human tutors [33] and we derive learning principles from both. Table 1 shows a summary of the principles that we found useful. The rest of this section describes these principles and discusses how they could be adopted in acquisition systems. More details on how current acquisition techniques are related to these principles are described in [20].

Instructional systems contain other components such as student models and domain models, but here we are focusing on tutoring principles and leave user modeling as future work.

Teaching/Learning principle	Tutoring literature
Introduce lesson topics and goals	Atlas-Andes, Meno-Tutor, Human tutorial dialog human learning
Use topics of the lesson as a guide	BE&E, UMFE
Subsumption to existing cognitive structure	human learning, WHY, Atlas-Andes
Immediate feedback	SOPHIE, Auto-Tutor, LISP tutor Human tutorial dialog, human learning
Generate educated guesses	Human tutorial dialog, QUADRATIC, PACT
Keep on track	GUIDON, SCHOLAR, TRAIN-Tutor
Indicate lack of understanding	Human tutorial dialog, WHY
Detect and fix "buggy" knowledge	SCHOLAR, Meno-Tutor, WHY, Buggy, CIRCSIM human learning
Learn deep models	PACT, Atlas-Andes
Learn domain language	Atlas-Andes, Meno-Tutor
Keep track of correct answers	Atlas-Andes
Prioritize learning tasks	WHY
Limit the nesting of the lesson to a handful	Atlas
Summarize what was learned	EXCHECK, TRAIN-Tutor, Meno-Tutor
Assess learned knowledge	WEST, Human tutorial dialog

Table 1. Some Tutoring and Learning Principles

References: Atlas [40], Atlas-Andes[37], BE&E [12], Buggy [6], CIRCSIM-tutor [44], EXCHECK [30], GUIDON [9], Human tutorial dialog [17], human learning [33, 19, 27, 3, 11, 15], LISP Tutor [2], Meno-Tutor [43], PACT [1], QUADRATIC [36], SCHOLAR [8], SOPHIE [5], TRAIN-Tutor [42], UMFE [38], WEST [7], WHY [39].

- **Introduce lesson topics and goals**
  In the beginning of the lesson, tutors often outline the topics to be learned during the session and try to assess the student's prior knowledge on these topics. For example, the advance organizer approach [3] lets the student see the big picture of what is to be learned and provides what the tutor's argument will be in order to bridge the gap between what the student may already know and what the student should learn. In educational systems, such as Meno-Tutor [43], as the tutor introduces general topics it asks exploratory questions in order to assess the student's prior knowledge. In fact, there are similar findings in teacher-student dialogs. Teachers often let students express how good or bad they are at given topic [17].

  Adopting the above tutoring principle, acquisition tools should start their dialogue by asking for the topic of the current lesson and establish assumed

as goals, strategies, or plans during the dialogue, or simply be taken into account in the design of the interaction.

prior knowledge. The topic of the lesson could be given as a set of terms to be defined, or a set of test problems that the system should be able to solve at the end of the lesson. Once the user specifies the topic, the system may assist the user to assess the current knowledge base in terms of the topic and bring up possibly relevant background knowledge. Missing prior knowledge can prompt a sub-dialogue for a background lesson.

– **Use topics of the lesson as a guide**

  In planning tutorial dialogues, instructional systems check what is being learned against the topics of the lesson [12] and try to avoid unfocused dialogue and digressions. In the process of learning, the terms brought up during the lesson are connected to the concepts learned [38].

  As in instructional systems, acquisition tools can use the topics of the lesson in checking how much progress the user made in building the knowledge base and in relating the terms introduced in the session to those topics.

– **Subsumption to existing cognitive structure**

  The subsumption theory by Ausubel [3] emphasizes that learning new material involves relating it to relevant ideas in the existing cognitive structure. The integration of new material with previous information can be done by analogies, generalizations and checking consistency. Through analogy, novel situations and problems can be understood in terms of familiar ones [19]. Effective human tutors ask for similarities and differences for similar cases [11]. In educational systems such as Atlas-Andes [37], the system points out differences between similar objects (e.g., speed vs. velocity) in terms of what they are and how they are calculated. Human tutors help students generalize when there are several similar cases [11]. For example, they suggest or point out the need to formulate a rule for similar cases by asking how the values of certain factors are related to the values of the dependent variables. Educational systems, such as Atlas [40], encourage students to abstract plans from the details to see the basic approach behind problem solving. Finally, cognitive dissonance theory [15] points out that people tend to seek consistency among their cognitions (i.e., beliefs, opinions). When there is an inconsistency (dissonance), something must change to eliminate the dissonance.

  Acquisition systems should follow this principle and assist users to: 1) learn new concepts from analogous concepts that already exist, 2) generalize definitions if similar things exist (and there could be plausible generalizations), and 3) make all new definitions consistent with existing knowledge.

– **Immediate feedback**

  Many educational systems provide immediate feedback on the quality of student's responses [5, 2]. The studies of feedback in a variety of instructional context find that immediate feedback is much more effective than feedback received after a delay [27]. Similarly, in the tutorial dialog study by Fox [17], tutors show immediate recognition of every step the student makes and their silence tends to presage the student's confusion. It is reported that in providing feedback, human tutors are more flexible than educational software, using high bandwidth communication to guide the students [33].

Based on this principle, we can make acquisition tools more actively involved in providing and obtaining feedback. For example, in addition to reporting how newly entered knowledge was understood and what errors were found, tools can ask for feedback on how results or answers being generated match the user's expectation.

- **Generate educated guesses**
  Some educational systems invite guesses on questions either in the process of letting the student discover the answers [36] or in the process of assessing the student's knowledge[1]. Likewise, in the studies of human tutoring, student often display their understanding by finishing the tutor's utterance and the tutor finds out what students understood by inviting their guesses (utterance completion strategy) [17].

  We can extend the existing capabilities of acquisition tools to provide educated guesses on how to fix problems based on their context. For example if there are salient features such as an action that can fix two errors at the same time, maybe it should be suggested as a most promising next step. If the guesses were wrong, it may be an indication of further missing knowledge and the system can show its surprise to the user and ask for further help.

- **Keep on track**
  If the student gives an incorrect answer, the tutor must immediately get the student back on track [8]. Some systems also detect change of directions [42] or check if the questions are irrelevant to the case at hand [9].

  Novice users of interactive acquisition interfaces often have difficulty in understanding if they are on the right track and if they are making progress [24]. We believe that acquisition interfaces should keep track of information regarding the progress made throughout the session and the tasks that remain to be addressed in the dialogue.

- **Indicate lack of understanding**
  Studies in human tutoring show cases where students themselves indicate lack of understanding of introduced terms [17], but tutors also point out the specific aspects that need to be understood by the student.[11].

  Some acquisition tools indicate to the user what is missing in the knowledge base [24] and users often use it to decide what to do next. Diagnosis questions should be useful to detect misunderstandings and missing knowledge.

- **Detect and fix "buggy" knowledge**
  Many educational systems have a tutoring goal of diagnosing the student's "bugs" [43, 39, 6] and question answering is often used in checking student's knowledge. However, simply telling that an error has occurred is much less useful than reminding the student of the current goal or pointing out a feature of the error [32]. If there are insufficient or unnecessary factors in a student's answers, experienced tutors pick counter examples to highlight the

problem [11]. In the process of checking, when the tutor does not understand the answer, sometimes the student is asked to rephrase the answer [8].

Most acquisition systems have a way of detecting errors and gaps in the knowledge base. However, as in the case of educational systems, instead of simply telling the errors found it is more useful to show the explanation of how and where the errors were found.

- **Learn deep models**

  The tutor and the student should focus on deep conceptual models and explanations rather than superficial ones [40]. Students should not only be expected to give the right answer but to do so for the right reasons. For example, when the student's answer is right, educational systems ask how the correct answer is generated [1, 40]. In some cases, to be able to ensure that the student understood the explanation educational systems use a set of check questions [37]. Studies of human tutoring show that students themselves occasionally try to check the reasoning behind the answers provided [17].

  Current acquisition tools do not have a good basis to evaluate or pursue depth in their knowledge base, though this is a long recognized shortcoming. One thing acquisition tools can do is to provide a way of enforcing users to check how the answers were generated and see if the system provides the right answer for the right reasons.

- **Learn domain language**

  Another interesting aspect of a lesson is learning to describe the new knowledge in terms that are appropriate in the domain at hand. Educators want to ensure that the students learn to talk science as a part of understanding of the science [40]. Teaching is more difficult when the student organizes and talks about knowledge in a different way than the tutor does [43].

  Acquiring domain language has not been a focus of knowledge base development in general. If an acquisition tool has a notion of checking the terms the users bring up in the process of entering knowledge, they can be highlighted to draw the user's attention.

- **Keep track of correct answers**

  Instructional systems keep track of the questions that the student is able to answer correctly as well as those answered incorrectly, which drives further interactions with the student. Some systems try more specific or simpler version of questions to keep better track of progress. [37].

  Some acquisition tools keep track of whether some set of test cases are answered correctly. However, they can be more actively used in guiding the acquisition dialog in terms of helping the user understand the current status of the knowledge base. For example, the acquisition tool can volunteer its own assessment of the kinds of questions that can be answered.

- **Prioritize learning tasks**

To handle multiple tasks and sub-tasks to be done, educational systems use priority rules. For example, systems can focus on errors before omissions and shorter fixes before longer fixes, prior steps before later steps, etc [11].

Similarly, some acquisition systems use a priority scheme to organize errors based on their type and the amount of help the system can provide.

- **Limit the nesting of the lesson**
  It appears that it is useful to limit the nesting of lessons to a handful [40], which seems it would help our acquisition tools keep track of what is going on as much as it helps a human student.
- **Summarize back to teacher what was learned**
  Many educational systems summarize the highlights at the end of the lesson [43, 30]. For example, EXCHECK prints out review of the proof for the student to give a clear picture of what has been done [30]. In some systems, when the tutor has given several hints, a summary may be given to ensure that the student has correct information just in case the student gave right answer by following hints without understanding the procedures [42].

  Acquisition tools do not actively provide a summary unless the user explicitly queries the knowledge base. Providing a summary of what has been learned in terms of the purpose of the lesson will be very useful for the user.
- **Assess learned knowledge**
  In their dialogs with human tutors, students often indicate how well they understand the topic as well as what has been learned [17]. Also some educational systems have a way of isolating the weaknesses in the student's knowledge and propose further lessons on those areas [7].

  Only some acquisition tools perform this kind of assessment. We believe that volunteering the assessment of how well the system understands certain topics will be very useful for the users.

4 Using Principles in Knowledge Acquisition

Based on our observations of teaching and learning principles described in the previous section, we are developing a system called SLICK[2]. The principles are used to steer the dialog with the user, and result in a more goal-oriented behavior that makes the system a more proactive learner.

We have designed SLICK as a front-end dialogue tool that can be layered over the functionality of existing acquisition interfaces. We are exploring the use of SLICK with SHAKEN [10], a tool that allows end users to specify process models in terms of their substeps and the objects involved, uses graphical input, and allows users to test the process model by asking questions and running a simulation. We are also using SLICK as a front-end dialogue tool for EXPECT [4], a tool that allows users to specify problem solving in terms of methods and

[2] Skills for Learning to Interactively Capture Knowledge

submethods, uses a structured editor for input, and allows users to pose both parameterized and instantiated problems for testing. In each case, the general learning principles described in this paper are operationalized by taking into account the features of the specific acquisition interface, in terms of the kinds of target knowledge they capture, the input modality offered to the user, and the testing and error checking strategies used. For example, the topic of the lesson in SHAKEN is a top-level process description and a set of objects that are involved in that process, while in EXPECT the topic of the lesson is given by a set of top-level problem solving goals. SLICK analyzes whether new terms introduced by the user relate to the topic of the lesson, checking this in SHAKEN by querying their appearance in the current expanded process description details and in EXPECT by checking their use in problem solving trees. We are also investigating how to include in SLICK useful dialogue management and user interaction techniques, as well as self-awareness capabilities that would enable it to assess the system's competence and confidence on the lesson topics as the dialogue with the user progresses.

5 Conclusion and Future Work

We have presented an analysis of instructional systems in terms of tutoring and learning principles and described how they could be useful in the context of interactive acquisition tools. We believe that they will play a central role in making acquisition tools proactive learners. We have started to incorporate these principles in our work and we are planning to perform user studies to collect feedback on the effectiveness of this addition.

Acknowledgments

This research was funded by the DARPA Rapid Knowledge Formation (RKF) program with award number N66001-00-C-8018. We would like to thank Ken Forbus, Lewis Johnson, Jeff Rickel, Paul Rosenbloom, David Traum, and Jim Blythe on their insightful comments on earlier drafts.

References

1. Aleven, V. & Koedinger, K. (2000). The need for tutorial dialog to support self-explanation. In *Proceedings of the AAAI Fall Symposium on Building Dialogue Systems for Tutorial Applications*.
2. Anderson, J. R., Conrad, F. G., & Corbett, A. T. (1989). Skill acquisition and the lisp tutor. *Cognitive Science*, 13:467–506.
3. Ausubel, D. (1968). *Educational psychology: A cognitive approach*. New York, Holt, Rinehart and Winston.
4. Blythe, J.; Kim, J.; Ramachandran, S.; and Gil, Y. (2001). An integrated environment for knowledge acquisition. In *Proceedings of the IUI-2001*.

5. Brown, J. S., Burton, R., & de Kleer, J. (1982). Pedagogical natural language and knowledge engineering techniques in SOPHIE I, II, III. In Derek, S. & Brown, J. S., (Eds.), *Intelligent Tutoring Systems*. New York, Academic Press.
6. Brown, J. S. & Burton, R. R. (1978). Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive Science*, 2:155–191.
7. Burton, R. & Brown, J. (1979). An investigation of computer coaching for informal learning activities. *International Journal of Man-Machine Studies*, 11:5–24.
8. Carbonell, J. R. (1970). AI in CAI: An artificial intelligence approach to computer-assisted instruction. *IEEE Transactions on Man-Machine Systems*, 11(4):190–202.
9. Clancey, W., (Ed.) (1987). *Knowledge-Based Tutoring:The GUIDON Program*. MIT press.
10. Clark, P., Thompson, J., Barker, K., Porter, B., Chaudhri, V., Rodriguez, A., Thomere, J., Mishra, S., Gil, Y., Hayes, P., & Reichherzer, T. (2001). Knowledge entry as the graphical assembly of components. In *Proceedings of K-CAP-2001*.
11. Collins, A. & Stevens, A. L. (1982). Goals and strategies of inquiry teachers. *Advances in Instructional Psychology*, 2:65–119.
12. Core, M. G., Moore, J. D., & Zinn, C. (2000). Supporting constructive learning with a feedback planner. In *Proceedings of the AAAI Fall Symposium on Building Dialogue Systems for Tutorial Applications*.
13. Davis, R. (1979). Interactive transfer of expertise: Acquisition of new inference rules. *Artificial Intelligence*, 12:121–157.
14. Eriksson, H., Shahar, Y., Tu, S. W., Puerta, A. R., & Musen, M. (1995). Task modeling with reusable problem-solving methods. *Artificial Intelligence*, 79:293–326.
15. Festinger, L. (1957). *A Theory of Cognitive Dissonance*. Stanford University Press.
16. Forbus, K. & Feltovich, P., (Eds.) (2001). *Smart Machines in Education*. AAAI press.
17. Fox, B. (1993). *The Human Tutorial Dialog Project*. Lawrence Erlbaum.
18. Gaines, B. R. & Shaw, M. (1993). Knowledge acquisition tools based on personal construct psychology. *The Knowledge Engineering Review*, 8(1):49–85.
19. Gentner, D., Holyoak, K. J., & Kokinov, B. N., (Eds.) (2001). *The analogical mind: Perspectives from cognitive science*. MIT press.
20. Gil, Y. & Kim, J. (2002). Interactive knowledge acquisition tools: A tutoring perspective. http://www.isi.edu/expect/papers/Interactive-KA-Tools-gil-kim-02.pdf (internal project report).
21. Gil, Y. & Melz, E. (1996). Explicit representations of problem-solving strategies to support knowledge acquisition. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*.
22. Ginsberg, A., Weiss, S., & Politakis, P. (1985). SEEK2: A generalized approach to automatic knowledge base refinement. In *Proceedings of IJCAI-85*.
23. Kim, J. & Gil, Y. (1999). Deriving expectations to guide knowledge base creation. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, pp. 235–241.
24. Kim, J. & Gil, Y. (2000). Acquiring problem-solving knowledge from end users: Putting interdependency models to the test. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*.
25. Kim, J. & Gil, Y. (2002). Proactive learning for interactive knowledge capture. http://www.isi.edu/expect/papers/KA-Dialog-Kim-Gil-02.pdf (internal project report).
26. Koedinger, K., Anderson, J., Hadley, W., & Mark, M. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, 8:30–43.

27. Kulik, J. & Kulik, C. (1988). Timing of feedback and verbal learning. *Review of Educational Research*, 58:79–97.
28. Lepper, M., Woolverton, M., Mumme, D., & Gurtner, J. (1993). Motivational techniques of expert human tutors: Lesson for the design of computer-based tutors. In Lajoie, S. & Derry, S., (Eds.), *Computers as Cognitive Tools*, pp. 75–105. Hillsdale.
29. Marcus, S. & McDermott, J. (1989). SALT: A knowledge acquisition language for propose-and-revise systems. *Artificial Intelligence*, 39(1):1–37.
30. McDonald, J. (1981). The EXCHECK CAI system. In Suppes, P., (Ed.), *University-level Computer-assisted Instruction at Stanford: 1968-1980*. Stanford.
31. McGuinness, D. L., Fikes, R., Rice, J., & Wilde, S. (2000). An environment for merging and testing large ontologies. In *Proceedings of KR-2000*.
32. McKendree, J. (1990). Effective feedback content for tutoring complex skills. *Human Computer Interactions*, 5:381–413.
33. Merrill, D. C., Reiser, B. J., Ranney, M., & Trafton, J. G. (1992). Effective tutoring techniques: A comparison of human tutors and intelligent tutoring systems. *The Journal of the Learning Sciences*, 2:277–305.
34. Murray, T. (1999). Authoring intelligent tutoring systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education*, 10:98–129.
35. Novak, J., (Ed.) (1998). *Learning, Creating, and Using Knowledge: Concept Maps as Facilitative Tools in Schools and Corporations*. Lawrence Erlbaum.
36. O'Shea, T. (1979). A self-improving Quadratic tutor. *International Journal of Man-Machine Studies*, 11:97–124.
37. Rose, C. P., Jordan, P., Ringenberg, M., Siler, S., VanLehn, K., & Weinstein, A. (2001). Interactive conceptual tutoring in Atlas-Andes. In *Proceedings of AI in Education*.
38. Sleeman, D. H. (1984). Inferring student models for intelligent computer-aided instruction. In Michalski, R. S., Carbonell, J. G., & Mitchell, T. M., (Eds.), *Machine Learning: An Artificial Intelligence Approach*, pp. 483–510. Springer.
39. Stevens, A. & Collins, A. (1977). The goal structure of a Socratic tutor. In *Proceedings of the National ACM Conference*.
40. VanLehn, K., Freedman, R., Pamela, J., Murray, C., Osan, R., Ringenberg, M., Rose, C., Schulze, K., Shelby, R., Treacy, D., Weinstein, A., & Wintersgill, M. (2000). Fading and deepening: The next steps for Andes and other model-tracing tutors. In *Proceedings of ITS-2000*.
41. Wenger, E., (Ed.) (1987). *Artificial Intelligence and Tutoring Systems*. Morgan Kaufmann.
42. Woolf, B. & Allen, J. (2000). Spoken language tutorial dialogue. In *Proceedings of the AAAI Fall Symposium on Building Dialogue Systems for Tutorial Applications*.
43. Woolf, B. P. & McDonald, D. D. (1984). Building a computer tutor: Design issues. *IEEE Computer*, 17(9):61–73.
44. Zhou, Y., Freedman, R., Michael, M. G. J., Rovick, A., & Evens, M. (1999). What should the tutor do when the student cannot answer a question? In *Proceedings of FLAIRS-99*.