Multiple Instructional Agents in an Intelligent Tutoring System[1]

Sherman R. Alpert[*] alpert@watson.ibm.com
Mark K. Singley[*] ksingley@watson.ibm.com
John M. Carroll[**] carroll@cs.vt.edu

In Proceedings of the AIED'99 Workshop on Animated and Personified Pedagogical Agents, July, 1999, Le Mans, Fance

[*]IBM T.J. Watson Research Center
POB 218
Yorktown Heights, NY 10598 USA

[**]Now at: Dept. of Computer Science
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061 USA

Abstract: We describe the Smalltalk Gurus[2], components of the MoleHill intelligent tutoring system for Smalltalk programming. The Gurus are tutorial agents personified in the user interface of MoleHill. They offer help on plans for achieving goals in the Smalltalk environment as well as remediation for students' incorrect and less-than-optimal plans. The Gurus' assistance is provided via the multimodal media of animation and voice-over audio. MoleHill employs multiple Gurus to deliver advice and instruction concerning disparate information domains.

THE GURU MODEL

Most computer users have had the experience of learning from a "guru."[2] We've wandered into the office of a local software expert seeking help and learned from this expert as she turned to her screen and simultaneously explained and demonstrated how to accomplish a computer-based task. On other occasions, we have had an expert watching over our shoulder as we performed a task on our computer and had the expert offer to "Let me show you a better way to do that." Such experiences demonstrate that observing an expert performing a task or having such an expert watch our own methods and comment on them are natural ways of learning (in this case, learning how to use computational systems; other researchers, e.g., Bannon, 1986, have reported the existence of this "local expert" learning model in computer-based settings). Such informal learning situations also mirror elements of the more traditional and formal apprenticeship instructional model for skill acquisition.

We have developed an elaborated computational version of the preceding model. Our *guru instructional model* is instantiated in MoleHill, an intelligent tutor for Smalltalk (Singley *et al*., 1991; 1993; Alpert *et al*., 1995). In MoleHill, personified instructional agents called the *Smalltalk Gurus* provide instructional support to novice Smalltalk programmers as they attempt to solve programming problems in an interactive development environment. Features of the guru instructional model as implemented by the Smalltalk Gurus include providing instruction in a mixed-initiative fashion–responding to user-requested help as well as offering unsolicited remediation for poor user plans, mirroring disjunctions in domain knowledge and skills by the use of multiple personified agents–one per subdomain, delivering instruction in the form of multimodal expert demonstrations, and offering students control over when tutoring is administered.

---

[1] This paper is largely excerpted from Alpert *et al*., 1995.

[2] According to Collins' Synonym Dictionary: *guru*: authority, master, mentor, teacher, tutor. The designation of "guru" for an expert is common in American computer culture.

MULTIPLE AGENTS

Domains of expertise often involve multiple subdomains of knowledge, information, and/or skills. In the case of Smalltalk programming tasks, skills involving use of the development environment are as important as use of the Smalltalk language itself. The environment provides a rich set of tools to facilitate the access, comprehension, and use of the extensive pre-existing code and structure which Smalltalk provides. In the long term, a programming environment that is so well integrated and supportive of its programming language is advantageous. However, its very richness adds to the seeming morass of information the novice Smalltalker is confronted with.

MoleHill's response is to provide help and remediation not only for code generated by users, but for their use of the environment as well—that is, tutoring not only on conceptual issues regarding object-oriented programming and the Smalltalk language but on procedural interface skills as well (e.g., how to use the functionality provided by the Smalltalk environment to locate a relevant piece of code to reuse). We thus have two distinct (albeit related) domains for which instruction may apply, namely (1) interface and navigational skills, and (2) language and conceptual issues regarding programming in Smalltalk. When instruction is offered, we wish to highlight which information domain is being addressed (cf. Kirson *et al*., 1988); we wish to reflect the categorical distinction in the to-be-learned material in the manner in which it is presented. The goal of this approach is to support the novice who is overwhelmed by the amount of information there is to digest when first encountering this new domain. As a number of behavioral studies have demonstrated, imposing a category-based organizational structure on instructional materials facilitates learner's assimilation of new information with related knowledge they already possess (Bower *et al*., 1969) and makes the information more memorable (e.g., Ornstein *et al*., 1974).

In MoleHill it is the Smalltalk Gurus who present information to apprentice programmers; hence, to accomplish the preceding instructional approach, MoleHill employs the organizational strategy of multiple gurus for multiple subdomains. In order to mirror the organizational cleft in the information to be delivered, MoleHill employs two Gurus: the Language Guru and the Interface Guru. Mapping the differentiated domains onto different agents should achieve our goal of making the domain distinction obvious. Learners need not begin to parse the message being presented by a Guru before assigning it to a conceptual category; this can be accomplished upfront based on which Guru is delivering the information, leaving cognitive resources to comprehension and assimilation of the presented material.

In order to clearly differentiate the Gurus from the user's perspective, the Gurus are personified using differing visual appearances (see Figures 1 and 2). They are differentiated further by the symbolic "emblems" on their shirts: the Interface Guru's shirt portrays windows of a graphical user interface while the Language Guru's shows the "Smalltalk balloon," a *de facto* logo for Smalltalk. Further, the Gurus have different voices matching their appearances—the Language Guru has a woman's voice, the Interface Guru a man's. In this way, the Gurus provide multiple *sensory cues* to facilitate users' organization of the information delivered by the Gurus.

MULTIMODAL AGENTS

The Smalltalk Gurus incorporate multiple media to offer help and remediation in a multimodal manner. Whereas simple text may suffice for the explication of some of this information the Gurus deliver, much of it is best conveyed by expert demonstration augmented by verbal explanation, elaboration, and justification. Again, this is what we see occurring naturally among gurus and novices in the workplace, and is how and why numerous skills have been traditionally taught in a mentor-apprentice instructional setting.

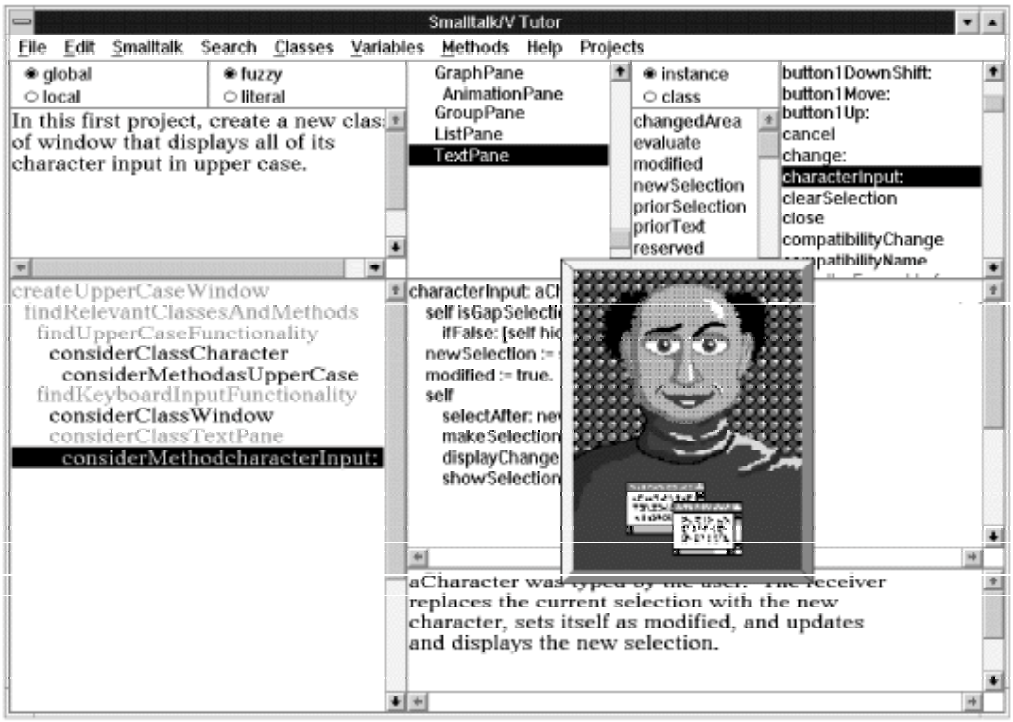

Figure 1. The MoleHill environment with the Inteface Guru visible. As the Guru "speaks" its facial features are animated. The GoalPoster appears in the lower left pane.

The Gurus provide bitmap animation (in-place, cartoon style animation of bitmap drawings), dynamic animation of the objects of the interface—the cursor, menus, windows, etc.—as if a user were controlling them with mouse and keyboard (what we call *interface animation*), and synchronized voice-over audio commentary. Via animated demonstrations accompanied by voice-overs, the Gurus interact with students to *tell* them *what* to do as well as to *show* them *how* to do it.

When either Guru is summoned by a user, it first appears as a "talking head" (see Figure 1) and introduces its forthcoming animated demonstration via voice audio and cartoon-like animation of its mouth, eyes, and eyebrows. For example, suppose the Interface Guru is called upon to provide help in finding a particular class in the Smalltalk class hierarchy browser. The Guru begins by saying "The fastest way to locate a specific class is to use the *Find Class* menu item. I'll show you how." The Guru then points to an area of the screen to direct the user's attention to where the demo will begin (as in Figure 2). Interface animation is then employed to convey the appearance and behavior of the interface as if a user were actually performing the task. For our example, the animation shows the cursor moving to the menu bar item labeled *Classes*, followed by the cursor changing to the image of a mouse with its left button highlighted so as to indicate a button press (see Figure 3), followed by the appearance of the pull-down menu itself, the cursor moving to the *Find Class* menu item, the left-button-down mouse image indicating a click on that item, and so on.

During these visual demonstrations, the Gurus' synchronized voice-overs provide commentary, explanation, elaboration, and/or justification for their animated actions. The Gurus use spoken, rather than written, text so as to avoid overloading of any one input channel and (as Palmiter and Elkerton (1991) point out as well) to avoid competition for attentional resources. Where appropriate, the Gurus' verbalizations make explicit the tacit knowledge involved in the process which is not visibly observable from the demo and would otherwise remain hidden from the learner (Brown *et al*., 1988; Collins *et al*., 1989). The audio commentary also provides a more abstracted summary view of the low-level, detailed process steps portrayed in the animation. In the case of the Interface Guru, for

example, several syntactic interface steps might be accompanied by a single comment which conveys the semantics of the actions being taken. For instance, the preceding animation segment, which involved several mouse movements onto specific interface objects as well as a number of button presses, would be accompanied by "First, select the *Find Class* item in the *Classes* menu."

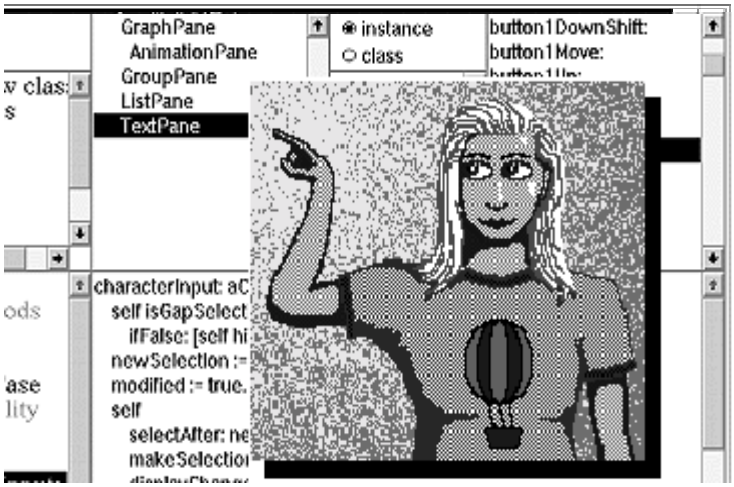

Figure 2. The Language Guru directing the user's attention to a particular portion of the screen via animation of "her" arm and eyes.

The range in abstraction level may be more pronounced with the Language Guru. This Guru often discusses a generalized conceptual issue while also talking the user through, and animating lower level details of, the implementation of a particular instantiation of the concept. Here, the two media are particularly complementary: the demonstrations provide concrete examples which should help users understand and remember the more generalized concepts and methods which the Guru's voice-over commentaries speak about (Collins *et al*., 1989); conversely, whereas the concrete examples are situated in particular programming tasks, the Guru's voice-overs attempt to speak at a level of generality which may support transfer to similar tasks (see also Sukaviriya *et al*., 1992).

Note that during interface animation, the Gurus do not portray the interface *exactly* as if a user were controlling the mouse and keyboard: as Payne *et al*. (1992) point out, the animation of interface objects being manipulated on the screen does not reveal details of the syntax of the actions (mouse clicks, double clicks, mouse movement with buttons held down, and so on) being performed by the "virtual user." There is also a problem of timing: if all activity were to take place in real-time, users attempting to learn from animated demos would not have enough time to read all menu selections, etc. (Payne *et al*., 1992) let alone process what was going on at anything but a superficial level. In order to avoid such difficulties, Guru demonstrations present a *stylized* representation of what the "user" is doing off-screen; for example, when, as described above, a click of the left mouse button is called for, the image of the cursor changes to the appearance of the mouse with the left button highlighted in red (see Figure 3).[3] In addition, the Gurus' interface animation includes time delays at the appropriate moments to allow the user to follow and digest the action.

One other point regarding Guru animations: originally, when the interface animation involved manipulating the mouse or cursor, or typing input, the Gurus' images changed accordingly to one in which they were, via animation, moving a mouse or typing at a keyboard. We subsequently moved away from this approach, leaving the interface animation as the single focal point for users' eyes and attention. We found that while animation can be useful, informative, and engaging, we must be careful not to offer visual activities that compete for the user's attention. Animation of a Guru at the

[3] This style is based on work on interface animation performed in our lab by Christine Sweeney (see Rosson *et al*., 1991). It is also similar to Sukaviriya's animated help systems (Sukaviriya *et al*., 1992, Sukaviriya & Foley, 1990).

mouse or keyboard concurrently with interface animation detracted from what we really wanted the user to focus on, namely the demonstration involving the animation of interface elements.

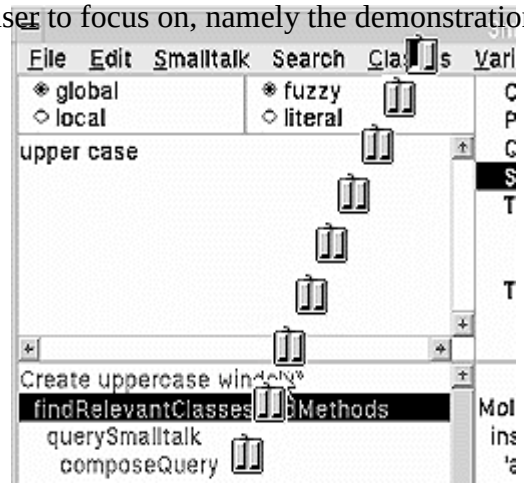

Figure 3. A "trace" of an interface animation segment showing the cursor being moved to a pulldown menu via the mouse (with no mouse button pressed) and the mouse's left button being pressed on the *Classes* menu. This animation continues to show the cursor being moved to the *Find Classes* menu item in the pulldown and the left button clicked again. Other interface manipulations animated during Guru demonstrations include keyboard input, animated by showing individual characters appearing one at a time as if being typed by a user.

ON THE PEDAGOGICAL USE OF ANIMATED DEMONSTRATIONS

Researchers have reported equivocal empirical results concerning the pedagogical effectiveness of animated demonstrations versus textual instructions for computer-based tasks (Palmiter & Elkerton, 1991; Palmiter *et al*., 1991); nonetheless, numerous other studies have demonstrated that users tend to ignore such text (e.g., Mack *et al*., 1983). Clearly then, comparing animated demos to text alone is meaningless if users refuse to make use of textual instructions. Other studies (Payne *et al*., 1992; Waterson & O'Malley, 1992) have in fact found a significant learning effect for animated demos for computer-based skills. Furthermore, Palmiter and Elkerton (1991) found that users preferred and even enjoyed animated demos and that text-only subjects complained they had difficulty *visualizing* the tasks they were being trained on. Demonstrations, on the other hand, supply the spatial knowledge and highlight the visual elements involved in the use of the computational environment (e.g., the appearance and location of particular menus, panes, buttons). Augmentation with explanatory spoken text which elaborates upon the low-level task details portrayed by the animation should further strengthen the efficacy of demonstrations.

An additional point regarding the pedagogical effectiveness of the Gurus' demonstrations concerns how situated they are in the process of performing real work. The Smalltalk Gurus do not provide demonstrations in a sterile, decontextualized introductory learning session, as in the aforementioned empirical studies: learners given explanations under such circumstances commonly have no idea how or when to make use of the provided information (Brown *et al*., 1988). Rather, Guru demos are provided in the situated milieu of actual problem solving, at which time the learner is most receptive to the ideas presented (Burton & Brown, 1982) and which should provide for more robust encoding of this knowledge (Anderson *et al.*, 1984).

GURU ROLES

The Gurus' roles in MoleHill include introducing new programming projects, responding to user-initiated help, delivering tutor-offered remediation of students' programming plans and use of the environment, and providing post-mortem wrapups when students complete projects. Here, we'll focus on the help and remediation task. (An elaborated version of this paper (Alpert *et al*., 1995) describes the other Guru roles). We explain how the user invokes the Gurus and motivate the use of the Gurus for this task.

The primary goal of the MoleHill tutor is to provide for a learning-by-doing experience by engaging users from the start in genuine tasks, and to support learners when they stumble into inevitable error situations or do not know how to proceed on a particular problem, consistent with the apprenticeship

and Minimalist (Carroll, 1990) instructional models. Assisting the user in such circumstances is the Gurus' primary role. One of MoleHill's components, the GoalPoster, provides students with a view of their goal-plan tree as they attempt to solve programming projects (Figure 1; see Singley *et al*., 1993, for a full description of this tool). At the start of a project, the GoalPoster is seeded with the top-level goal, that is, the project's intended result (e.g., "Create uppercase window" in Figure 1). The GoalPoster then dynamically displays the tutor's ongoing interpretation of the student's actions in terms of inferred subgoals and plans. These goals and plans thus become not only reified for the student, but the source of potential dialog between tutor and student.

For example, students may request help at any time for any GoalPoster entry. In response to help requests, the tutor employs a repetition heuristic. The first time help for a particular goal entry is solicited, the tutor's response is provided in textual form. The text aims to provide the user with information as to the goal's role in the current project and/or a simple hint as to how to attack the goal. Subsequent help requests for the same goal are answered with greater specificity regarding an appropriate plan for that goal. Whereas some goals lend themselves to simple textual help, others are ultimately best tutored by demonstration. Thus, depending on the content of what is to be explained, the more specific help for some goals involves invoking the appropriate Guru to provide an instructive demonstration.

In addition to user-requested help, the tutor may *offer* help or remediation to the student. In attacking a project subgoal, a student may employ a plan which is faulty (i.e., one which will not lead to a correct solution) or one that arrives at a working solution but which represents a less than optimal approach. In such situations, the appropriate Guru (more than likely, the Language Guru) informs the student that it has something to say regarding the student's current activities. It does so by annotating the corresponding GoalPoster entry with a miniature Guru icon (see Figure 4). At his or her own initiative, the user may ask to receive Guru comments and demonstrations for any such annotated goal in the GoalPoster. This user request may occur at any time after the Guru leaves behind its iconic "calling card" on the GoalPoster entry.

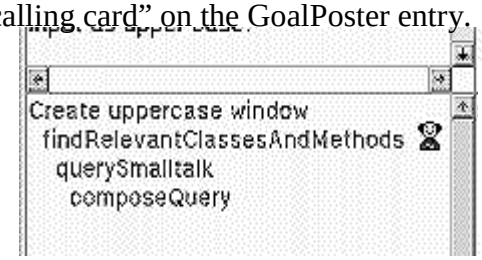

Figure 4. The Guru's iconic "calling card:" A GoalPoster entry annotated with a Guru icon, informing the user that a Guru is offering remediation for that goal

Thus, the Gurus do not *force* the learner to view their remediation. Although many have advocated immediate feedback as errors occur (e.g., Anderson *et al*., 1984) others such as McKendree *et al*. (1992) have determined empirically that such intervention can be unwanted and annoying. The latter researchers found that students repeatedly expressed the desire to repair errors before receiving imposed help. Also, advisory interruptions may cause users to forget their currently active plan (Carroll & Aaronson, 1988). We wanted to have the tutor be flexible enough to allow learners to decide for themselves whether or not to interrupt their work. We, therefore, inform students that a Guru has something to say, but leave it up to them to ask the Guru to speak. Leaving control with learners regarding whether and when to interrupt their ongoing work is another aspect of the guru instructional model.

SUMMARY

The obvious drawback of apprenticeship learning or informal instruction from gurus in the workplace is that not all learners are fortunate enough to have access to such mentors. The Smalltalk Gurus provide such access in a computer-based form. As a result, users may learn computer-based skills in

an environment which incorporates many of the advantages of these instructional models. The computational Gurus, in fact, may offer advantages over human mentors in that students are supported without the angst of being watched by someone who may wield power over them (Carroll, 1994). Students are thus able to learn in a non-threatening environment in which mistakes are not met with the disapproval of an impatient teacher or judgmental authority figure. Further, in our computer-based version, students have a greater sense of control in that *they* decide when to permit mentor interruptions. Lastly, multiple personified advisors may offer an advantage over a single tutor if their multiplicity successfully mirrors and conveys the categorical distinctions in the tutored domain.

MoleHill's Gurus instantiate what we have called the guru instructional model. This model incorporates a number of ideas from the intelligent tutoring community and others unique to the Smalltalk Gurus. Features of the model as implemented in MoleHill include the following; items (c) through (e) apply specifically to the manner in which the personified interface agents are utilized: (a) supplying proactive and reactive instructional support, the latter by "observing" student work and offering advice for not only faulty but sub-optimal student plans; (b) providing students a sense of empowerment by allowing them to decide whether and when to interrupt their work for guru advice and commentary; (c) presenting instructional support in the form of multimodal expert demonstrations and providing for complementary levels of abstraction and explanation among the media involved; (d) providing an organization for the presentation of tutorial advice which makes salient the split between disparate knowledge or information categories, allowing users to more easily sort out which category is being addressed and thereby facilitating the integration of new information with related knowledge; and (e) manifesting and highlighting this organizational split by supplying multiple, specialized, personified advisors, one for each of the distinct domains, and providing multiple sensory cues (both visual and auditory) to further highlight this split.

Lastly, we feel that the Gurus' appearance, animations, and use of multiple media add a sense of engagement to the tutor while fulfilling the serious role of helping, remediating, and supporting the learner. Engaged learners should be motivated learners, willing to spend more time on task, which can significantly enhance learning.

References

Alpert, S.R., Singley, M.K., & Carroll, J.M. (1995). Multiple multimodal mentors: delivering computer-based instruction via specialized anthropomorphic advisors. *Behaviour & Information Technology*, *14*(2), 69-79.

Anderson, J.R., Boyle, C.F., Farrell, R., & Reiser, B.J. (1989). Cognitive principles in the design of computer tutors. In P. Morris (Ed.), *Modelling Cognition*. NY: Wiley.

Bannon, L.J. (1986). Helping users help each other, In D. Norman and S.W. Draper (Eds.), *User Centered System* (pp. 339-410). Hillsdale, NJ: Erlbaum.

Bower, G.H., Clark, M.C., Lesgold, A.M., & Winzenz, D. (1969). Hierarchical retrieval schemes in recall of categorized word lists. *Journal of Verbal Learning and Verbal Behavior, 8,* 323-343.

Brown, J.S., Collins, A., & Duguid, P. (1988). *Cognitive apprenticeship, situated cognition, and social interaction*. Institute for Research on Learning Report No. IRL88-0008, June, 1988. Also Bolt Beranek and Newman Research Report 6886.

Burton, R.R. & Brown, J.S. (1982). An investigation of computer coaching for informal learning activities. In D. Sleeman & J.S. Brown (Eds.), *Intelligent Tutoring Systems* (pp. 79-98). London: Academic.

Carroll, J.M. (1990). *The Nurnberg Funnel: Designing Minimalist Instruction for Practical Computer Skill*. Cambridge, MA: MIT Press.

Carroll, J.M. (1994) Making *use* a design representation, *Communications of the ACM*, *37* (12), 28-35.

Carroll, J.M. & Aaronson, A.P. (1988). Learning by doing with simulated intelligent help. *Communications of the ACM*, *31*, 1064-1079.

Collins, A., Brown, J.S., & Newman, S.E. (1989). Cognitive Apprenticeship: Teaching the crafts of reading, writing, and mathematics. In L.B. Resnick (Ed.), *Knowing, Learning, and Instruction: Essays in Honor Of Robert Glaser* (pp. 453-494). Hillsdale, NJ: Lawrence Erlbaum.

Kirson, D.S., Carroll, J.M., Eckhoff, R., Kelley, J.F. and Canetti, S. (1988). Specialist help with multiple media. *SIGGRAPH Video Review, #30*. (shown at CHI'88 Conference on Human Factors in Computing Systems).

Mack, R.L., Lewis, C., & Carroll, J.M. (1983). Learning to use office systems: Problems and prospects, *ACM Transcations on Office Information Systems*, *1*, 254-271.

McKendree, J., Radlinski, B., & Atwood, M.E. (1992). The Grace tutor: A qualified success, In C. Frasson, G. Gautier, & G.I. McCalla (Eds.), *Intelligent Tutoring Systems: Second International Conference, ITS '92 Proceedings* (pp. 677-684). Berlin: Springer-Verlag.

Ornstein, P.A., Trabasso, T., & Johnson-Laird, P.N. (1974). To organize is to remember: The effects of instruction to organize and recall, *Journal of Experimental Psychology*, *103*, 1014-1018.

Palmiter, S. & Elkerton, J. (1991). An evaluation of animated demonstrations for learning computer-based tasks. In S.P. Robertson, G.M. Olson, & J.S. Olson (eds.), *CHI'91 Conference Proceedings*, pp. 257-263.

Palmiter, S., Elkerton, J., & Baggett, P. (1991). Animated demonstrations vs written instructions for learning procedural tasks: A preliminary investigation, *International Journal of Man-Machine Studies*, *34*, 687-701.

Payne, S.J., Chesworth, L., & Hill, E. (1992). Animated demonstrations for exploratory learners, *Interacting with Computers*, *4* (1), 3-22.

Rosson, M.B., Carroll, J.M., & Sweeney, C. (1991). A View Matcher for reusing Smalltalk classes. In S.P. Robertson, G.M. Olson, & J.S. Olson (Eds.), *CHI'91 Conference Proceedings* (pp. 277-283). NY: ACM.

Singley, M.K., Carroll, J.M., & Alpert, S.R. (1991). Psychological design rationale for an intelligent tutoring system for Smalltalk. In J. Koenemann-Belliveau, T.G. Moher, & S.P. Robertson (Eds.), *Empirical Studies of Programmers: Fourth Workshop* (pp. 196-209). Norwood, NJ: Ablex.

Singley, M.K., Carroll, J.M., & Alpert, S.R. (1993). Incidental reification of goals in an intelligent tutor for Smalltalk, In E. Lemut, B. du Boulay, & G. Dettori (Eds.), *Cognitive Models and Intelligent Environments for Learning Programming* (pp. 145-155). New York: Springer-Verlag.

Sukaviriya, P.N., Isaacs, E., & Bharat, K. (1992). Multimedia help: A prototype and an experiment, In P. Bauersfeld, J. Bennett, & G. Lynch (Eds.), *CHI'92 Conference Proceedings* (pp. 433-434). NY: ACM.

Sukaviriya, P.N. & Foley, J.D. (1990). Coupling a UI framework with automatic generation of context-sensitive animated help, In *Proceedings of the ACM SIGGRAPH Symposium on User Interface Software and Technology (UIST'90)* (pp. 152-166). NY: ACM.

Waterson, P. & O'Malley, C. (1992). Using animated demonstrations to teach graphics skills. In A. Monk, D. Diaper, and M.D. Harrison (Eds.), *People and Computers VII: Proceedings of HCI'92* (pp. 463-474). Cambridge, UK: Cambridge University Press.