DOCUMENT RESUME

ED 332 688	IR 015 015
AUTHOR	Russell, Daniel M.; Kelley, Loretta
TITLE	Using IDE in Instructional Design: Encouraging Reflective Instruction Design Through Automated Design Tools.
PUB DATE	91
NOTE	10p.; Paper presented at the Annual Conference of the American Educational Research Association (Chicago, IL, April 3-7, 1991).
PUB TYPE	Speeches/Conference Papers (150)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	Computer Assisted Instruction; Course Organization; *Curriculum Development; Feedback; Higher Education; High Schools; Hypermedia; *Instructional Design; Mathematics Instruction; *Participative Decision Making; Preservice Teacher Education
IDENTIFIERS	Stanford Teacher Education Program

ABSTRACT

The Instructional Design Environment (IDE), a computer assisted instruct on tool for instructional design, has been incorporated into the curriculum and instructional development in mathematics instruction in the Stanford Teacher Education Program (STEP). (STEP is a 12-month program leading to an M.A. in education which emphasizes content focus and reflective and collaborative teaching.) Previously used as an industrial instructional analysis and development tool, IDE has now been used as a tool for teacher training. It uses hypermedia-style representations of subject matter and an instructional design model that students use to develop curriculum and organize courses. The results of using IDE include: (1) students are prompted to follow the instructor's design guidelines and standards; (2) continual feedback between students and instructor can be provided; (3) participative decision making between students is promoted; (4) reflection among students as to the effects of their decision is promoted; and (5) exploration of subject matter arrangements is possible. Students using IDE for curriculum development in mathematics are willing to explore alternative lesson organizations, have a deeper understanding of the subject matter, and think more about instructional design practice. An example of an instructional model, a lesson card, a rationale card, and a subtopic card are appended. (4 references) (DB)

Using IDE in Instructional Design: Encouraging Reflective Instruction Design Through Automated Design Tools

by

Daniel M. Russell and Loretta Kelley

Paper presented at AERA, 1991.

Using IDE in instructional design: Encouraging reflective instruction design through automated design tools

Daniel M. Russell
Institute for Research on Learning &
Xerox Palo Alto Research Center
Palo Alto, CA 94304

Loretta Kelley
Stanford University
Stanford Teacher Education Program
Stanford, CA 94305

Abstract: IDE, a computer-based analysis tool for instructional design, has been incorporated into the Curriculum and Instruction in Mathematics course in the Stanford Teach Education Program (STEP). Previously used as an industrial instructional analysis and development tool, here we report on the first use of IDE as a tool for teacher training. We describe IDE, STEP and the purpose of the course, then we discuss the effects that IDE had on the students' understanding of instructional design and their subject matter.

IDE: The Instructional Design Environment (IDE) is a computer system under development at the Institute for Research on Learning (IRL) that supports the task of instructional design. IDE users create hypermedia-style representations of the subject matter and of the course under construction in order to analyze both what is to be taught, and how to teach it. Running on a common platform (Macintosh) in a networked environment, IDE also supports group collaboration by allowing designers to share individual and group work easily over the network and by supporting an individual's analysis of materials within the overall design. A central feature of IDE is that the instructional design model, with its supporting analyses and representations, may be changed as subject matter or circumstances change. An instructional design defines the course structure and methodology in IDE, which the student then uses to actually create curriculum by selecting and organizing materials according to this structure.

The initial objective in building IDE was simply to facilitate some of the onerous tasks involved in instruction design, and to provide a mechanism for recording rationales of instructional design decisions. However, in our early observations of IDE use in industrial uses, we noted that IDE appeared to have a greater effect on its users than we anticipated. The belief that these effects would be even more beneficial to those learning to plan instruction suggested that we use IDE in teacher education.

STEP is a 12-month program leading to a M.A. in education and a California secondary teaching credential. As a program, STEP emphasizes content focus, reflective and collaborative teaching. During the summer of 1990, we incorporated IDE into the mathematics Curriculum and Instruction class because our observations led us to believe that IDE would support these goals during the teaching of curriculum development.

Goals of Project : By using IDE at STEP, we hoped to discover if IDE actually supported these objectives. It is important to have student users *understand* the function and process of curriculum development (and its tradeoffs) and to have experience that can be used in their professional lives to aid their design practice. In addition, we want to give the students a structure in which to organize their subject matter content and that will simultaneously communicate an instructional design process to the students.

Procedure: IDE was used in the summer term of STEP's mathematics C&I class. Eleven students were enrolled in this intense eight-week course. The instructor designed the framework of an entire high school algebra course, and completed one topic as a model within that framework. Each student was assigned a topic and expected to use IDE to develop the topic as a complete set of lesson plans (approximately 20 lesson plans) including possible assessment questions, anticipated learner difficulties with the particular content, and a rationale for curricular decisions. The class's objective was to collaborate in creating a complete algebra course. At the end of the class, each student will receive a copy of the completed course which can be used in subsequent teaching.

Methods: To characterize the students using IDE, we initially assessed the students' background with a questionaire to determine previous teaching and computing experience. After each work session with IDE, the student were asked to complete a short report describing their use of IDE, with problems and benefits experienced. As the students worked, field observation notes were taken about student work practices with IDE. We solicited feedback in the classroom and, at the completion of the 8-week class, we performed videotaped, structured interviews to assess the way in which IDE-using students plan their instruction and understand their subject matter. Based on this information, we made a set of observations about the effects of using IDE. In addition, we will contrast the unit plans created IDE with analogous work performed by students from last year's mathematics Curriculum and Instruction course as taught by one of us (L. Kelley).

Results: We noted effects on both the instructor and the students. *[A] IDE allowed the instructor to encourage top-down, incremental design practice by the students.* The way IDE represents information about the course, and that way it is structured, prompts the student to follow the instructor's design guidelines and standards. *[B IDE allowed continual feedback to the instructor about the status of each individual's work.* Since the students did all of their work on the system, and since their work could be integrated together and shared over the network, it was simple to provide rapid turnaround of feedback from the instructor to the students. *[C] Students became reflective about the effects of their design decisions.* In the IDE design model created by the STEP instructor, explaining (rationalizing) important design decisions was a central part of the way development work progressed. By requiring that the students rationalize their decisions, we found that fewer decisions were made implicitly, and that previously implicit choices were made explicit. *[D] Collaboration between students was enhanced.* We noted three important causes: First, IDE "browsers" (a graphical display of the current course structure and organization) give the student an ability to see what work they are doing, and how it fits into the overall work of the entire class. Second, because other work was easily available in the joint project database, students could model their work by examining other materials in the complete system. Third, In prior classes, work was done by students asynchronously and independently, offering little opportunity (or need) to interact. Since IDE allows work to proceed in parallel, and since the lessons must integrate by the end of the term and were integrated weekly, the computer lab became a focus for group work and collaboration about both IDE use and subject matter discussions. *[E] Students could explore various arrangements of the subject matter.* In IDE, the representation of course materials is chunked into manipulable fragments with connections between fragments explicitly available. Students, using browsers and hypermedia navigation techniques, can easily rearrange, restructure, and resequence curriculum components to explore (and discover) alternative course designs.

Perspective / Theoretical Framework: There is an emerging field of instructional design support systems. Systems such as IDioM [Gustafson & Reeves, 90], IDExpert

[Merrill, Li, Jones, 90], and the system of [Jones & Wipond, 90] all provide automated tools to assist in analysis, design and development of instructional materials. In contrast to IDE, these systems are committed to a pre-determined instructional design paradigm without the ability to support work in other design models or with representations that are tuned to a specific subject matter. In addition, collaboration is only beginning emerge as a theme in these computer-aided instructional design systems, making their use for teacher instruction in group work settings problematic.

Summary: In this use of IDE as a tool for teacher training, we have found that students using IDE for curriculum development in mathematics are willing to explore alternative lesson organizations, have deepened their understanding of the subject matter, and have become reflective in their thinking about instructional design practice.

References:

[Gustafson & Reeves, 90]
Gustafson, K., Reeves, T.
"IDioM: A platform for a course development expert system"
Educational Technology, v XXX, n 3, (March, 1990)

[Merrill, Li & Jones, 90]
Merrill, M. S., Li, Z., Jones, M.
"The second generation instructional design research program"
Educational Technology, v XXX, n 3, (March, 1990)

[Jones & Wipond, 90]
Jones, M., Wipond, K.
"Curriculum and knowledge representation in a knowledge-based system for curriculum development"
Educational Technology, v XXX, n 3, (March, 1990)

[Russell, et al., 90]
Russell, D. M., Burton, R., Jordan, D., Jensen, A., Rogers, R., Cohen, J.
"Creating instruction with IDE: Tools for instructional designers"
Intelligent Tutoring Media, v 1, n 1, (May, 1990)

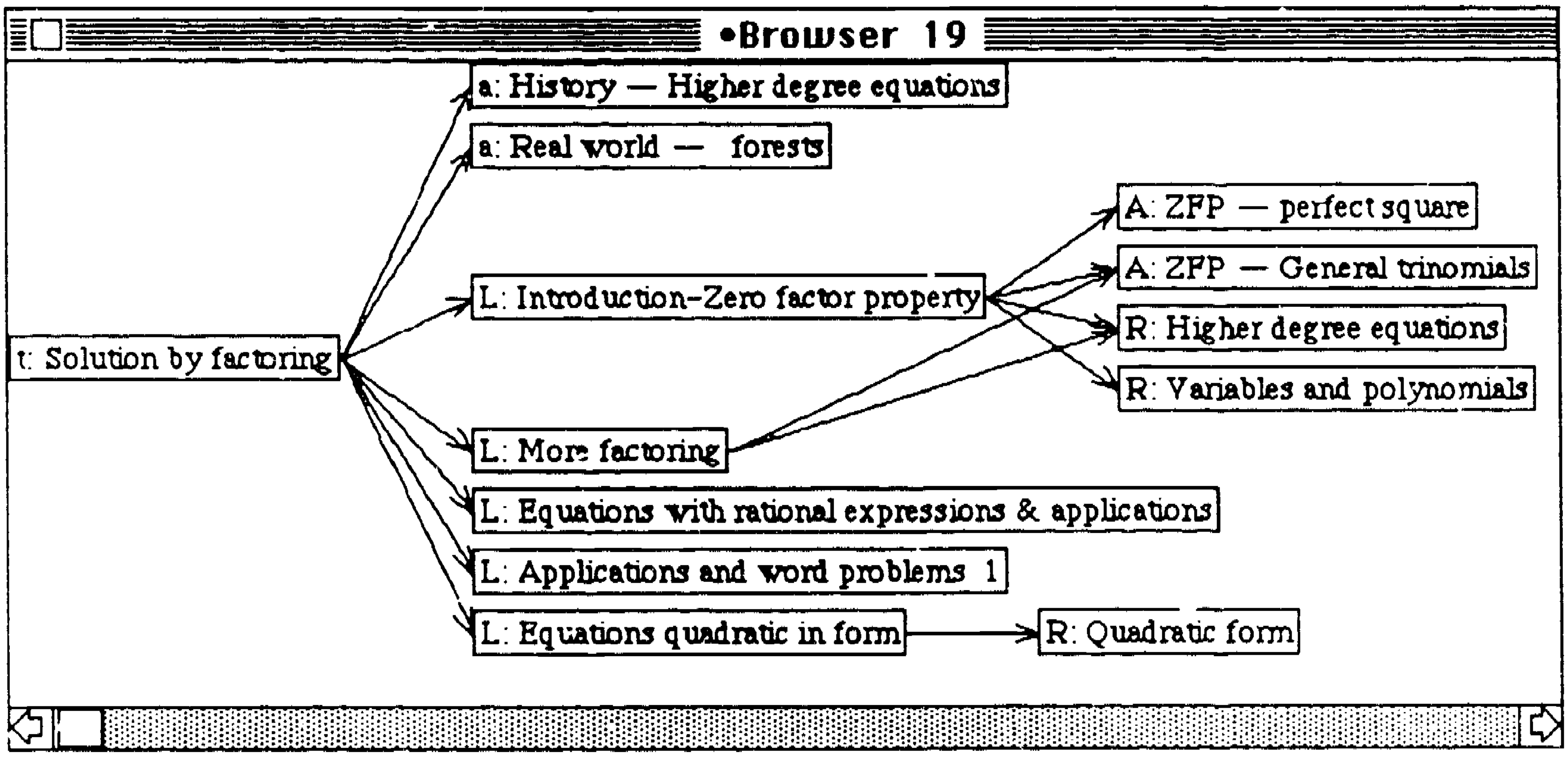

Figure 1: This browser shows the cards created by a STEP student using IDE to design a set of Lessons (L) to teach the subtopic (t) "Solution by factoring." Using browsers, the structure of the course can be easily seen and manipulated. This particular instructional organization (topics leading to subtopics that are organized into lessons, assessments (A), application (a) sections and rationales (R)) was created by the Curriculum and Instruction class instructor. This is simply one of a large variety of instructional models that can be implemented in IDE.

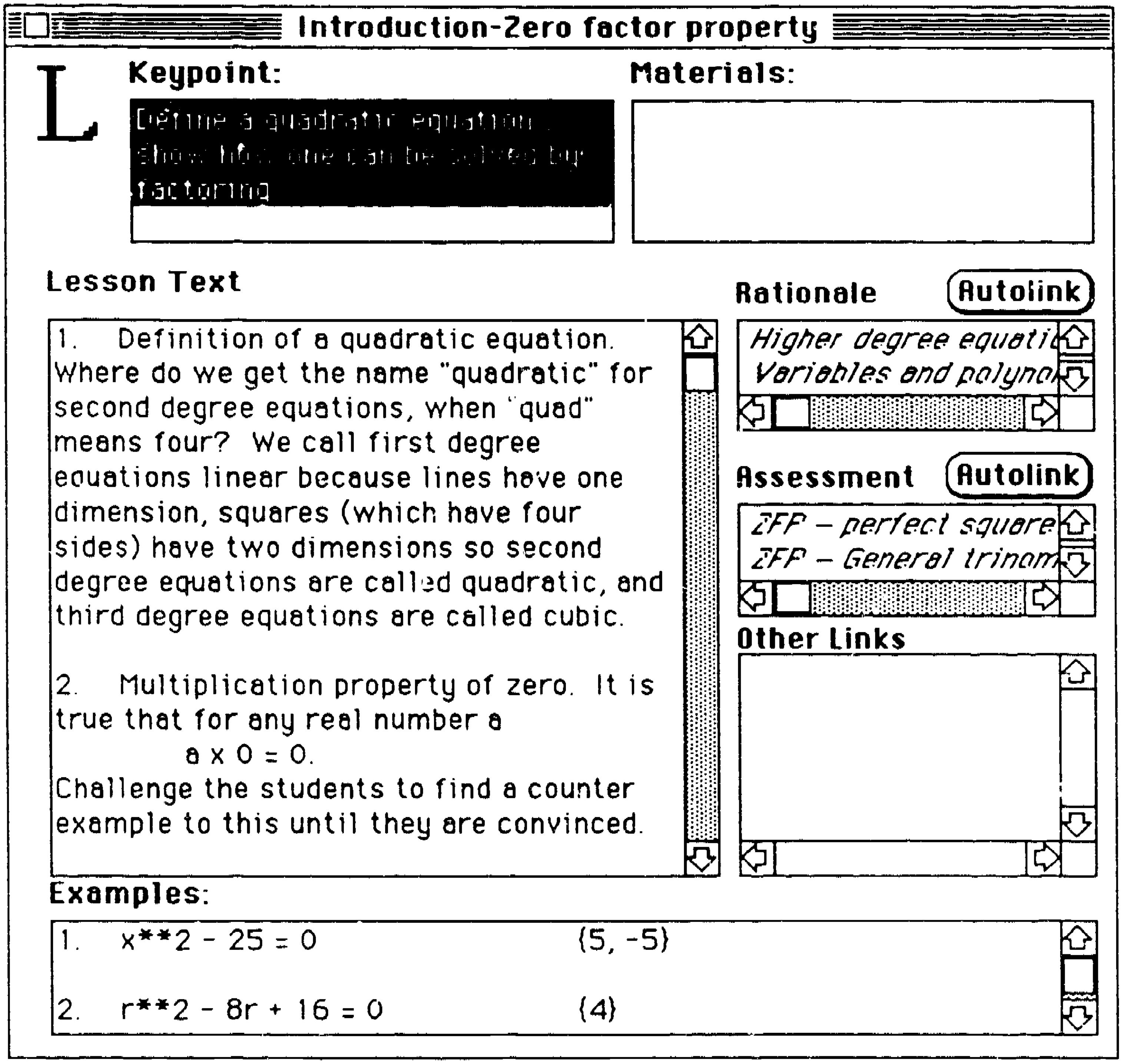

Figure 2: A Lesson card contains the text of the lesson plan to teach a subconcept(s) within the overall course. Each region on this card (e.g., Lesson Text, or Examples) is a field that is to be filled in by the course designer. Note that some field have text: the "Keypoint" is a summary of what essential idea is contained in this lesson. Some fields, such as "Rationale" or "Assessment" are links to other cards which carry the rationale statements and assessment instruments for this lesson.

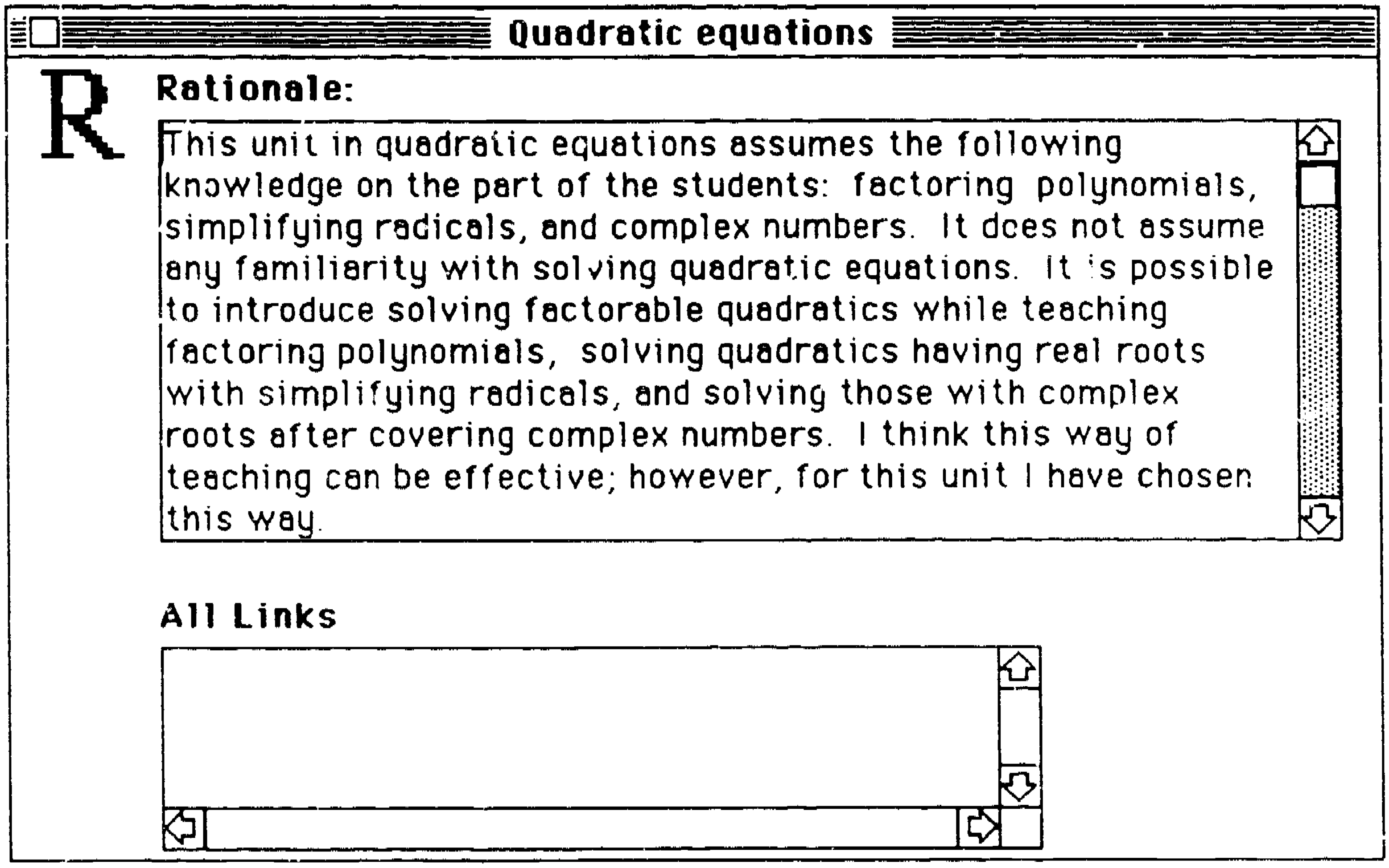

Figure 3: A Rationale card is a place where the course designers can annotate the course with arguments (pro or con) for particular design decisions. Requiring designers to include argumentation (and references!) for design decisions heightens an increased awareness of the tradeoffs being made during design, and opens up the design process to collaborative discussion. For teacher training, rationales require entirely new ways of looking at the source materials (in this case, high school algebra) and incorporating that perspective into course design and development.

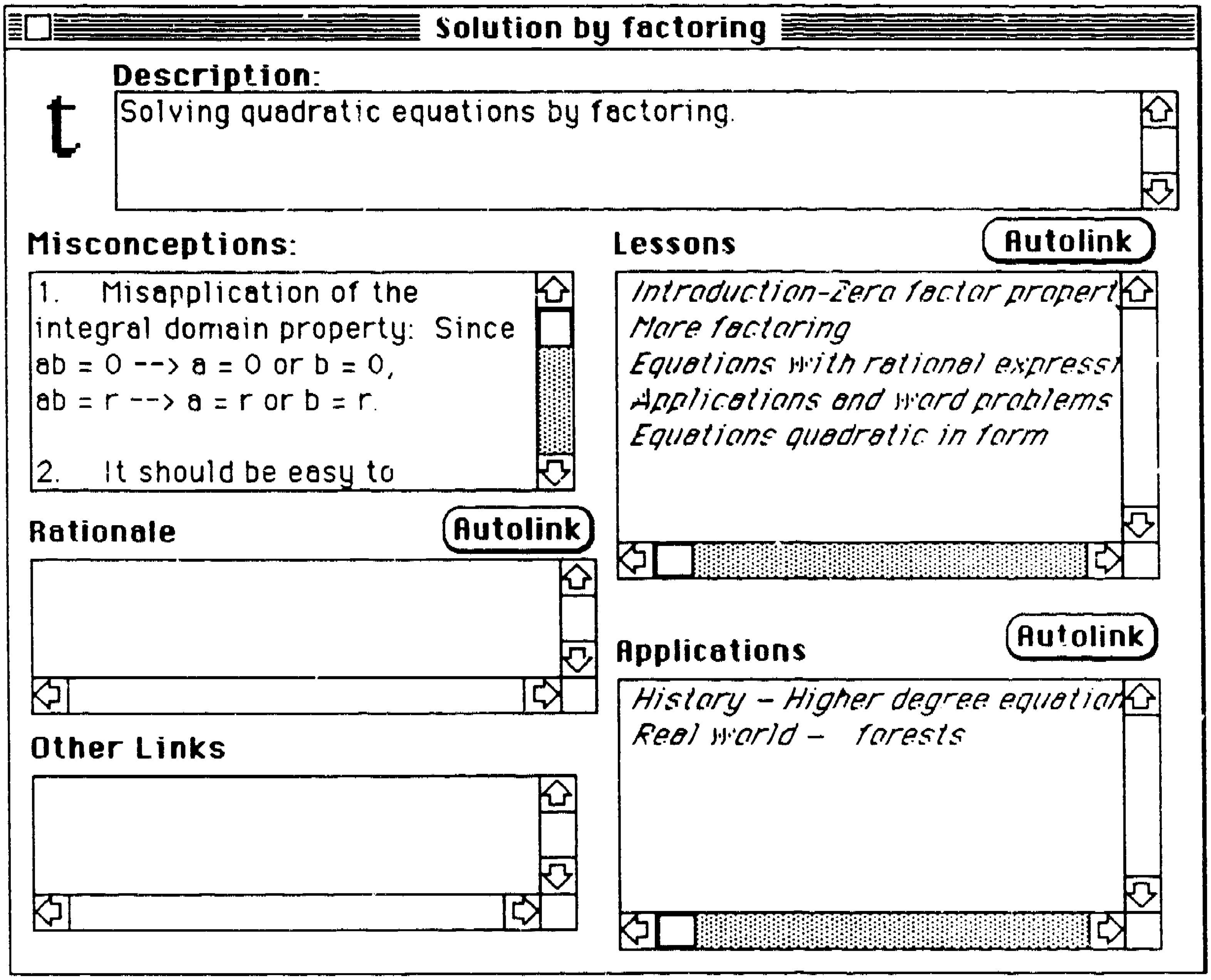

Figure 4: The subtopic card acts as an organizer for the lessons, rationale and applications. Here, it is clear that no rationale statements have been completed. The empty field acts as a gentle reminder to the student that to complete the design, an argument for the lesson organization must be included.