# Discourse Control for Tutoring: Case Studies in Example Generation

Beverly Park Woolf
Dan Suthers
Tom Murray

COINS Technical Report 88-49

# Discourse Control for Tutoring:

# Case Studies in Example Generation[1]

Beverly Woolf
Dan Suthers
Tom Murray

Department of Computer and Information Science,
University of Massachusetts,
Amherst, Massachusetts 01003

## Abstract

We describe two automated tutors based on a model of tutorial discourse as a process of controlled movement through spaces of topics, examples, and responses. The selection and modification of examples using a domain independent example generator is discussed in some detail.

**Keywords:** AI in Education

[1]This work was supported in part by a grant from the National Science Foundation, Materials Development Research, No. 8751362. Support was also received from Air Force Systems Command, Rome Air Development Center, Griffith AFB, New York, 13441 and the Air Force Office of Scientific Research, Bolling AFB, DC 20332 under contract No. F30602-85-C-0008. This contract supports the Northeast Artificial Intelligence Consortium (NAIC). Additional support was received from an ONR University Research Initiative Contract No. N00014-86-K-0764 and from an Apple Corporation grant.

# 1 Tutoring Processes

Effective tutoring requires dynamic reasoning about selection of machine responses as well as careful coordination of machine utterances. It requires a more fluid dialogue than has been required for question and answer or summarization systems. Tutoring requires genuine flexibility in the choice of topics, presentations, examples, and responses.

In order to achieve such flexible selection of topics and machine responses, a tutoring system should reason about abstract directives, such as:

> "When the student is *confused*, present a *less complex* example."

To do this, the machine should be able to decide when the student appears confused, and then know how to construct a less complex example by weakening features along a specific dimension. This reasoning is to be contrasted with the use of situation-specific rules such as "If the student answers the question correctly, present example #32." This latter kind of specification, frequently used in conventional teaching systems, is responsive only to local information, i.e., to which problem was presented and how the student responded. It ignores situation abstractions derived from a sequence of interactions. It also forces the programmer into a tedious level of detail.

In the past, networks and production rule formalisms have been used to identify admissible instructional strategies and examples (e.g., Cerri & Breuker, [1981]). Typically these systems were domain-dependent and restricted to a narrow set of didactic responses. An exception to this was Clancey's GUIDON system [1982] which defined a large set of domain-independent rules to guide a tutor in reasoning about the student's knowledge and alternative dialogues. Fundamental to our perspective is the view that tutoring conversation and the presentation of examples or feedback are motivated by general rules and principles of tutoring and discourse. Such rules are currently being researched by many investigators, including ourselves [Murray et al., submitted; Brown et al., 1986; Grosz & Sidner, 1985; Reichman, 1985; van Lehn, 1983]. Cognitive results from such work are used to inform the processing models we build.

We have developed a three-layered control architecture for dynamic control of discourse at three levels of granularity. Control in these three layers is

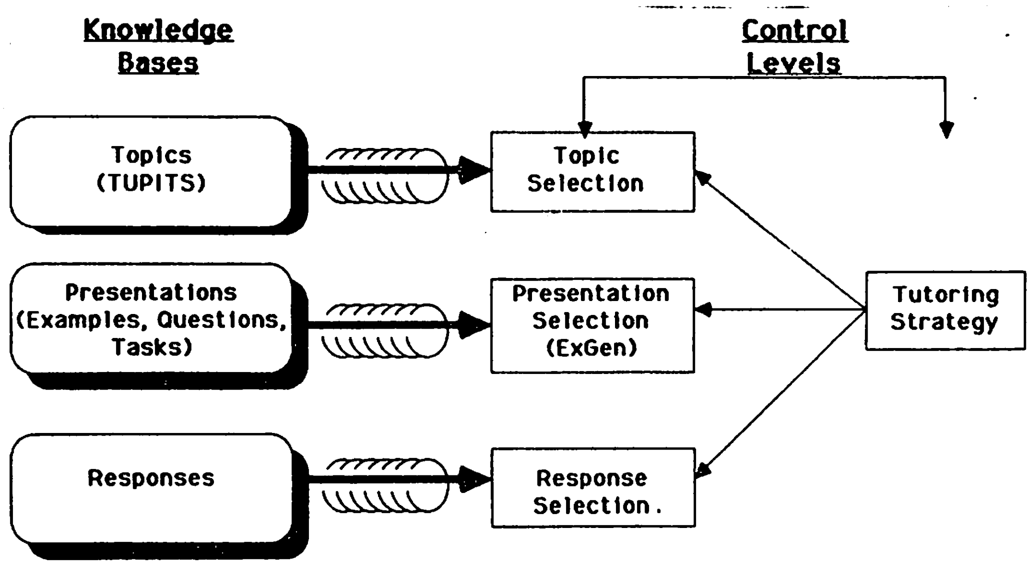



Figure 1: Three (Plus One) Layers of Control

sensitive to the current tutoring strategy in ways to be discussed. Selection of a tutoring strategy constitutes a form of meta-control, as it impacts on how other control decisions are made. The present paper sketches this architecture, indicating how a given strategy manifests itself in the three layers of control. The decisions made at the middle level of presentation selection are discussed in more detail, illustrating use of the example generation tool named ExGen.

## 2 Layers of Discourse Control

The three-layered discourse control architecture is depicted in Figure 1. The system is able to show flexibility in its choice of topic, presentation, and response. The fourth control component to be added is the choice of tutoring strategy which parameterizes how the choices in the other three layers are made.

The Knowledge Unit level of control makes decisions about which topic to discuss, concept to tutor, or misconception to remediate. The Presentation level will then instantiate the chosen topic into various ways of interacting with the student, such as giving examples of a concept, asking questions, providing descriptions and definitions, and so on. The Response level will interpret the chosen presentation, call the user interface to execute it, and generate informative and motivational feedback according to the tutoring strategy. The interactions are recorded in the student and discourse models and used to drive diagonsis of misconceptions.

Our tutors are responsive to the strategy which is currently selected, and we are beginning to implement automated selection of tutoring strategies. Each strategy specifies the values for various control parameters at each of the three levels of control. In our implemented systems, changing control behavior is simply a matter of changing which such strategy is designated as current. We are ready to address the more difficult aspect of implementing strategic meta-control: knowing *when* to change the current strategy.

The three levels of control are described briefly in this section in terms of the choices available at each level, and how the current strategy impacts on these choices. In the next section, the choice of presentation (examples, questions, descriptions, definitions, etc.) is described in more detail.

## 2.1 Topic Selection

Presently we represent topics within a Knowledge Unit network of topic frames which explicitly expresses relationships between topics such as prerequisites, corequisites, and related misconceptions [Woolf et al., 1988]. An important notion about the network is that it is declarative; it contains a structured space of concepts, but does not mandate any particular order for traversal of this space.

The current tutoring strategy manifests itself in the control at this level by parameterizing the algorithm used to traverse the Knowledge Unit network based on classifications of and relations between these units. Several major strategies have thus far been implemented. The tutor might always teach prerequisites before teaching the goal topic. Alternatively, the tutor might provide a diagnostic probe to see if the student knows a topic. If the student doesn't exhibit enough knowledge on the probe, then prerequisites might be presented. These prerequisites may be reached in various ways, such as depth-first and breadth-first traversal. An intermediate strategy which we have used is to specialize the prerequisite relation into "hard" prerequisites, which are always covered before the goal topic, and "soft" prerequisites, taught only when the student displays a deficiency.

There are also "Mis-Knowledge Units," which represent common misconceptions or knowledge "bugs," and ways to remediate them. These are inserted opportunistically into the discourse. The tutoring strategy parameterizes this aspect of Knowledge Unit selection by indicating whether such remediation should occur as soon as the misconception is suspected, or wait until the current Knowledge Unit has been completed.

## 2.2 Presentation Selection

The intermediate level of discourse control utilizes an example generation tool called ExGen [Suthers & Rissland, submitted]. Examples, questions, and descriptions of the concept being taught are all treated as "examples" by ExGen. A "seed" example base contains prototypical presentations of each type. ExGen's modification routine expands this into a much larger virtual space of presentations as needed. The goal is to enable the tutor to have flexibility in its presentation of examples and questions/tasks that accompany those examples, without the knowledge engineering burden of representing all possible presentations explicitly.

ExGen takes a list of weighted constraints called *requests*, and returns an example. The constraints are written in a language which allows one to describe logical combinations of the desired attributes of the example, and the weights on them represent the relative importance of each of these attributes. The returned example generally meets as many of the constraints as possible in the priority order indicated by the weights.

In the tutors described here, we embed ExGen within a presentation generation module which is tailored to teach physics concepts. This module is driven by *example generation specialists*, or knowledge sources, each of which examines the current discourse and student models and produces requests (weighted constraints) to be given to ExGen. These example generation specialists may be thought of as tutoring rules, encoding such general prescriptives as "when starting a new topic, give a start-up example," or "ask questions requiring a qualitative response before those involving quantities." The particular specialists we use are described in Section 3.3.

The tutoring strategy impacts on this middle layer of control by prioritizing the relative importance of the recommendations produced by each of the example generation specialists. Within a strategy, each specialist has a weight which is multiplied by the weight of the requests produced by the specialists. Altering the behavior of the Presentation control is simply a matter of changing the weights on the specialists by selecting a new strategy.

For instance, the specialists include one which requests that presentations describing the current Knowledge Unit be given, and another requesting that the student be questioned. These competing requests are ordered by the current tutoring strategy. We are also examining strategies for temporal ordering of the presentation of examples, such as Bridging Analogies and Incremental Generalization (Section 3.3).

## 2.3 Response Selection

We have begun to represent alternative responses to student actions. The choices available at this level are concerned with how much information to give the student in the interactive presentation, and what motivational comments to make, for example whether or not to:

- tell the student *whether* she is correct;
- tell the student *what* the correct response is;
- tell the student *why* her response is correct or incorrect;
- give hints or leading questions;
- challenge the student's answer with a counter-suggestion; or
- provide additional information which extends the content of the current question-answer interaction.

Motivational responses may include encouragement, congratulations, challenges, and other statements with affective or prelocutionary content. As before, the current tutoring strategy specifies which of these feedback and follow-up responses will be generated. The strategy may also specify that they be predicated on whether the student's response was correct, or even that no response is to be given.

## 2.4 Coordinating the Control Levels

The choices we have described leave us with the problem of determining what combinations of control parameters for each level are most appropriate. For example, at each level there are the extremes of a verbose, spoon-feeding tutor vs. one that is conservative about giving away information and interacts Socratically with questions. At the Knowledge Unit level, we could always cover prerequisites before goal concepts vs. only selecting prerequisites if the student fails to make progress in the goal concept. At the Presentation level, we could always give definitions and descriptions first vs. presenting examples and asking questions to get the student to think about the current concept before volunteering any information to her. At the Response level, we could always tell the student whether she was correct, what the correct response is, and why, vs. not saying any thing at all, relying on the next presentation to stimulate the student into knowing whether she was right or wrong.

If a tutoring strategy specified the more verbose extreme at all three levels, we'd have a chatty tutor that never let the student think for herself. On the other hand, a strategy which was information-stingy at all three levels would likely frustrate the student. This may suggest that a tutor should take the middle road at each level of control. However, it may also mean that these extremes should be mixed across levels. For instance, a strategy for filling in the gaps in a student's knowledge of an area she has already encountered could use the probe-first method at the Knowledge Unit level, and thus only cover a topic if the student displays a lack of understanding of it, but utilized more informative finer-grained interactions. A Socratic approach to teaching a domain new to the student could do nearly the opposite, always starting with prerequisites before goal concepts, but initiating each concept with questions about that concept rather than giving away the definition. The completion of the meta-control component of our tutoring systems will enable us to explore these issues further.

# 3 Generating Examples for Presentation Selection

In this section, the Presentation layer of our architecture is described in greater detail. We first provide a sample of examples generated by this layer and then focus on the application-independent ExGen mechanism, and the pedagogical feature-dimensions and the example-generation specialists which select appropriate presentations.

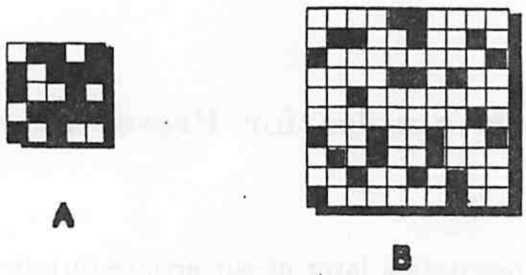
## 3.1 Samples from the ExGen System

Several examples from the thermodynamics and statics tutors are presented in Figure 2. ExGen places an example on the screen along with an accompanying question or problem. For each tutor, examples can be generated that differ only slightly or perhaps only along a single dimension from a prior example. ExGen can present substantially more complex (or more rich or more simple) examples based on the perceived needs of the student, as explained in the next sections.

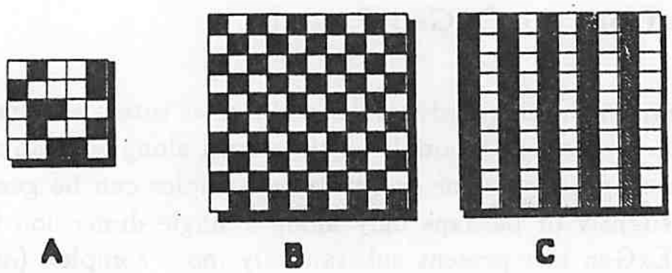
## 3.2 The ExGen Kernel

The application-independent mechanisms of ExGen are located within the ExGen Kernel. This mechanism parses, prioritizes, and performs certain optimizations on the list of requests made by specialists, and oversees the retrieval and modification of selected examples. These processes are summarized below (see Suthers and Rissland [submitted] for more detail).

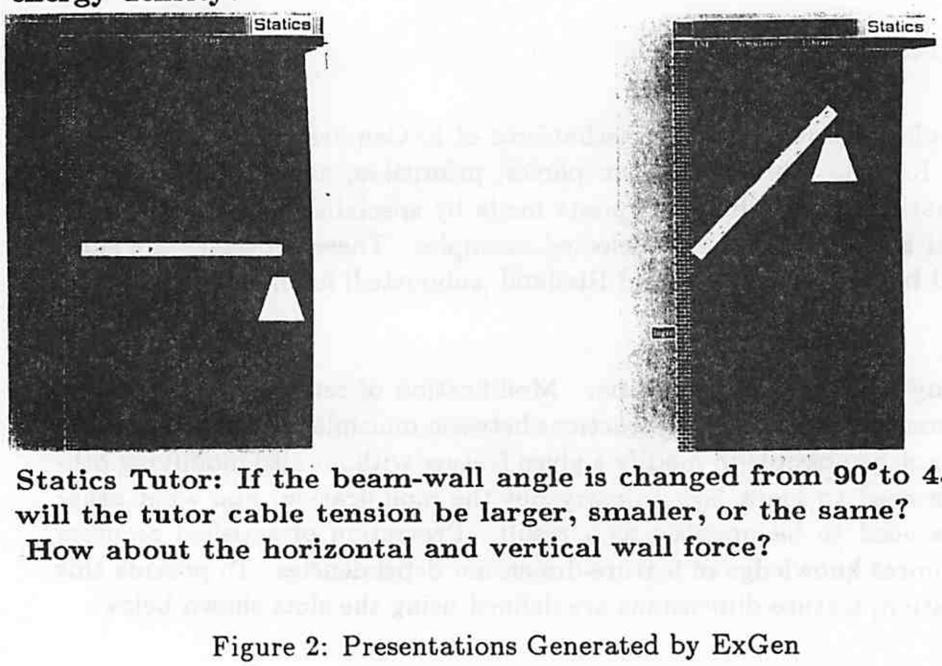
**Defining Feature-Dimensions.** Modification of retrieved examples requires consideration of the interactions between multiple feature-dimensions. It may not be possible to modify a given feature without also modifying others. We need to know how to carry out the modification, and what other features need to be updated as a result. Protection of satisfied requests also requires knowledge of feature-dimension dependencies. To provide this information, feature-dimensions are defined using the slots shown below:




**Thermodynamics Tutor: Which universe has greater energy?**
**Which universe has greater energy *density*?**




**B: Tutor: Compare each universe pairwise. Which has greater energy density?**




**Statics Tutor: If the beam-wall angle is changed from 90°to 45°, will the tutor cable tension be larger, smaller, or the same? How about the horizontal and vertical wall force?**

Figure 2: Presentations Generated by ExGen

**If-Needed:** A function which maps an example to the value of the feature dimension for the example. It is used to determine values for derived feature-dimensions, typically by examining the values on some other feature-dimensions.

**To-Modify:** A function which constitutes the specific knowledge of how to modify examples. The `to-modify` methods of derived feature-dimensions are written in terms of recursive modification of their underlying primitive features.

**Depends-On:** A list of the feature-dimensions of which the feature being defined is an abstraction. This is identical to those referenced by the `if-needed` and `to-modify` methods of the feature being defined. From `depends-on`, ExGen computes two relations used in the Modifier: `protect`, used for goal protection and `update`, used for propagation of changed values.

**Sample-Values:** A list of some sample values for the feature-dimension, which allows ExGen to extend the virtual example base beyond that possible by closure of all combinations of the attributes of existing examples. This aids in knowledge engineering by greatly reducing the number of "seed" examples needed for a particular application.

We have analyzed the interdependencies of the thermodynamics and statics feature-dimensions, and have identified those which are *primary*, in that they cannot be computed from other features, and those which are *derived*, in that they can. Some of these dimensions are primarily descriptive of the physical characteristics of the example presented, while others indicate the pedagogical characteristics of the presentation. The latter are generally independent of the tutoring domain. Some pedagogical feature-dimensions are described in the next section.

**Example Generation.** The example generator operates within the context of a tutor that maintains the student and discourse models. The example selection specialists are sensitive to these dynamic models. Example generation begins by allowing the specialists to post their requests, which are then prioritized, resulting in a list of requests. The requests are converted into disjunctive normal form where possible, and certain optimizing

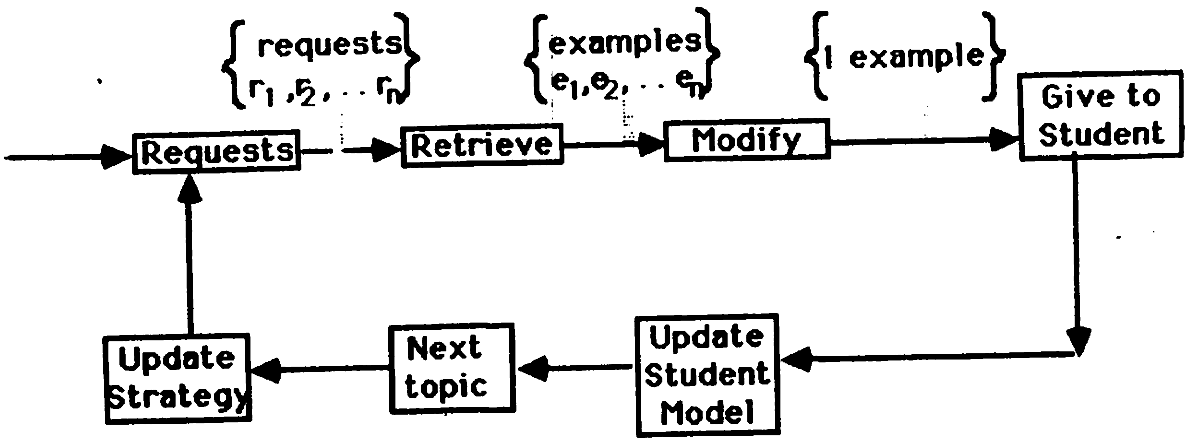



Figure 3: Example Generation Mechanism

manipulations are performed on the request list (see Suthers and Rissland [submitted]).

When given the requests, the Retriever returns the set of examples which satisfies as many of the requests as possible in the order given, without letting the retrieved set go empty. One of the (equivalent) retrieved examples is then chosen as the working example (see Figure 3).

The job of the Modifier is to take the working example and the normalized list of requests, and attempt to satisfy those requests which were not satisfied by retrieval, where it can be done without violating requests of higher priority. This is done by iterating over the requests in priority order, keeping a `protected-features` list, which consists of the union of the `protect` images of the features whose values satisfy requests which have already been checked. The `update` lists are used to recompute derived features as needed to maintain network consistency, as in a constraint propagation network. The modifier code itself does not contain any application dependent knowledge about the feature-dimensions or their interdependencies: this resides in the definitions of the features.

Modification allows us to tailor examples to the particulars of the current situation, filling out an example base of "seed" examples provided by the human expert as required to meet the needs of an individual student. If the example was modified, the new example is recorded in the example base. This saves processing if similar situations are encountered in the future. Then the example is returned to be interpreted by the user interface.

## 3.3 Pedagogy of Example Selection

Examples available to ExGen are described in terms of attributes relevant to tutoring in that subject. A sampling of the pedagogical feature-dimensions used by ExGen and the specialists which use them to select the next presentation are detailed below.

**Pedagogical Feature Dimensions.** These describe the pedagogical quality of the task that will be requested of the student. They provide a way for the tutor to reason about what kind of presentation to make and what kind of task to supply. These features are the same for both statics and thermodynamics and we expect them to be improved on as we refine current tutors and build new ones.

**KUs-Questioned:** Those Knowledge Units which the presentation asks the student about.

**KUs-Described:** Those Knowledge Units which the presentation gives a definition or description of.

**KUs-Assumed:** Those Knowledge Units which the presentation assumes the student understands, in that terminology associated with them is used without definition.

**Response-Type:** Whether the student simply acknowledges the presentation, gives a multiple-choice response, is asked to enter some particular quantity or name, or to interact with a simulated environment.

**Response-Quality:** An ordered dimension indicating the quality of information the student is asked to give, including no response, judging the existence of an entity, comparing two entities, giving a qualitative description, and giving a quantitative value.

**Specialists.** Example selection specialists test the state of the student and discourse models and then make a judgment about which example, question, or description to present next. Each specialist represents one (possibly complex) rule-like heuristic. These specialists are the result of cognitive studies of expert tutors.

While all specialists are actively generating requests at all times, their requests are combined and weighed against each other in an attempt to produce an optimally "satisficing" presentation according to the current tutorial strategy. Many specialists have local state, enabling them to change their recommendations as appropriate over time or to be sensitive to their previous behavior. A sampling of specialists currently used in the two tutors is shown below:

**Response-Quality-Progression:** When a Knowledge Unit is first introduced, prefers simple qualitative questions, such as acknowledging the existence of an entity. Progresses to prefer comparison questions, and then more complex qualitative and quantitative responses. This represents the tactic of requiring responses of increasing informational content.

**Startup-New-KU:** When a Knowledge Unit is started for the first time, requests an example classified as a startup example.

**Taxonomic-Coverage:** For each Knowledge Unit, asks for coverage of the taxonomic-classes of examples (e.g., startup, reference, counter, and extreme examples [Rissland (Michner), 1978]). The requests increase in strength as more examples are given for the Knowledge Unit without covering the desired classes.

**Describe-Current-KU:** Requests that the selected presentation explicitly define or describe the current concept.

**Question-Current-KU:** Requests that the selected presentation ask questions to gauge the student's understanding of the current concept.

**Variety:** Requests that certain features be varied, with increasing strength as a feature is repeated without change. This is essentially an anti-boredom consideration.

Two of our most useful example selection strategies are **Incremental Generalization** and **Bridging Analogies**. Bridging analogies [Brown et al., 1986; Murray et al., submitted] is used when a student has already exhibited misunderstanding of a concept, determined by presenting a relatively difficult case of a concept. Given evidence of a misunderstanding the tutor introduces carefully chosen examples in an effort to teach the concept. First the tutor finds an anchor, or simple case that the student seems to comprehend. Then it presents examples that repeatedly "split the difference" between the most difficult example of the concept the student *understands*, and the simplest example of the concept that the student *misunderstands*. The tutor explicitly asks the student to compare and contrast these two examples, thus engendering a sense of cognitive dissonance until the student sees the similarity in the two examples and changes his mind about the misconceived concept.

Incremental generalization is a selection strategy that differs from the bridging analogy strategy by working "forward" from an easily understood "start-up example" [Michner (Rissland), 1978] example of a concept. It attempts to generalize the student's understanding of the concept to include qualitative variants of the original example of the concept along sub-ranges of feature-dimensions that are *irrelevant* to the concept being generalized. The strategy is to present feature values that are "closer" to those of the start-up example first, working gradually to the extreme values, thus allowing the student to incrementally generalize the concept. For instance, while teaching energy density, the thermodynamics tutor varies the energy pattern from randomly distributed energy, through uniformly distributed but nonrandom patterns, to the extreme case where all the energy is in one region of the system.

Bridging is intended for tutoring situations in which anchors familiar to the student exist, and where the student is already familiar with the feature-dimensions involved. Incremental generalization is intended for domains whose concepts are new to the student. This completes the discussion of the pedagogical feature-dimensions and specialists.

## 4 Conclusions

Tutoring requires sophisticated and dynamic reasoning about selection of machine response. We have implemented an architecture for tutoring discourse control that selects and modifies tutoring elements at the level of topics, examples, and responses. The architecture enables the machine to produce incrementally more complex or more simple examples based upon the tutor's reasoning about the student's previous work. We suggest that tutoring control may profitably be seen as control of movement through spaces of tutoring elements at each of these levels.

With the exception of automated selection of tutoring strategies, all the mechanisms described above have been implemented for both the Statics and the Thermodynamics tutor. We will soon be ready to test the effectiveness of tutoring strategies expressed as parameters controlling the way discourse decisions are made at the three levels discussed. Further understanding of the situations in which each strategy is most effective will eventually enable us to build the component for strategic meta-control.

## 5 References


Ashley K., Knowing What to Ask Next and Why: Asking Pertinents Questions Using Cases and Hypotheticals, *Proceedings of the Cognitive Science Society*, Lawrence Erlbaum Associates, Hillsdale, NJ, 1986.

Ashley K., and Rissland, E., Explaining and Arguing with Examples, *Proceedings of the National Association of Artificial Intelligence*, Morgan Kaufmann, Inc., Los Altos, CA, 1984.

Brown, D., Clement, J., and Murray, T., Tutoring Specifications for a Computer Program Which Uses Analogies to Teach Mechanics, *Cognitive Processes Research Group Working Paper*, Department of Physics, University of Massachusetts, Amherst, MA, April 1986.

Cerri, S. A., and Breuker, J., A Rather Intelligent Language Teacher, in Hart (Ed.), *Studies in Language Learning*, University of Illinois, Urbana, IL, Vol. 3, # 1, pp. 182-192, 1981.

Clancey, W., Tutoring rules for Guiding a Case Method Dialogue, in D.

Sleeman and J. S. Brown (Eds.), *Intelligent Tutoring Systems*, Academic Press, Cambridge, MA, 1982.

Grosz, B., and Sidner, C., The Structures of Discourse Structure, *Proceedings of the National Association of Artificial Intelligence*, Morgan Kaufmann, Los Altos, CA, 1985.

Michner (Rissland), E., Understanding Understanding Mathematics, *Cognitive Science*, Vol. 2, 4, 1978.

Murray, T., Brown, T., Clement, J., and Schultz, K., Using Analogies to Bridge Concepts, submitted to *The Journal of Artificial Intelligence and Education*.

Reichman, R., *Making Computers Talk Like You and Me*, MIT Press, 1985.

Suthers, D., and Rissland, E., Constraint Manipulation for Example Generation, *Proceedings of the National Association of Artificial Intelligence*, Morgan Kaufmann, Los Altos, CA, submitted 1988.

van Lehn, K., Felicity Conditions for Human Skill Acquisition: Validating an AI Theory, *Report Number CSL-21*, Palo Alto, CA, Xerox Palo Alto Research Center, 1983.

Woolf, B., Murray, T., Suthers D., and Schultz, K., Primitive Knowledge Units for Tutoring Systems, in *Proceedings of the International Conference on Intelligent Tutoring Systems (ITS88)*, June, 1988.

Woolf, B., and Murray. T., A Framework for Representing Tutorial Discourse *Proceedings of the International Joint Conference on Artificial Intelligence*, Milan, Italy 1987.

Woolf, B., and McDonald, D., Context-dependent Transitions in Tutoring Discourse, *Proceedings of the National Association of Artificial Intelligence*, Morgan Kaufmann, Los Altos, CA, 1984.