A Revision-Based Model of Instructional Multi-Paragraph Discourse Production

Proceedings of the Annual Meeting of the Cognitive Science Society, 13(0)

Lester, James C.
Porter, Bruce W.

1991

A Revision-Based Model of Instructional Multi-Paragraph Discourse Production*

James C. Lester
(lester@cs.utexas.edu)

Bruce W. Porter
(porter@cs.utexas.edu)

Artificial Intelligence Laboratory
Department of Computer Sciences
University of Texas at Austin
Austin, Texas 78712

Abstract

To communicate effectively, intelligent tutoring systems should be able to generate clear explanations of phenomena in their domain. To explain complex phenomena in scientific domains, they must be able to produce extensive multi-paragraph discourse. Traditionally, discourse planners have taken a *monotonic* approach to generation: once they make a decision, that decision is never revoked. Because these approaches make no provision for evaluating and modifying a plan after it has been constructed, their flexibility is limited. This inflexibility is particularly acute when attempting to generate multi-paragraph discourse.

We propose a *revision-based* model of discourse planning that constructs instructional multi-paragraph discourse plans, evaluates them, and restructures them. This is accomplished by delaying organizational commitments as long as possible and interleaving the planner's content determination and organization activities. This model accords well with research on writing. It has been implemented in an experimental system, KNIGHT, a discourse generator for intelligent tutoring systems. KNIGHT generates multi-paragraph explanations in the domain of biology. A domain expert has analyzed KNIGHT's explanations and found them to be clear and accurate.

Introduction

The principle mission of an intelligent tutoring system (ITS) is to communicate knowledge to a student [Wenger, 1987]. The explanation generator of an ITS should generate explanations that are appropriate for the student's current mastery of domain concepts. Furthermore, an ITS for scientific domains must be able to explain very complex phenomena, so its explanation generator must be able to produce extensive (multi-paragraph) explanations.

For the past two decades, researchers in natural language generation have been exploring the problem of constructing text from a formal representation of knowledge. Generators that produce multi-sentential or multi-paragraph text have traditionally taken a *monotonic* approach to discourse planning. [1] We say that a discourse planner is monotonic if, once it makes a decision, that decision is never revoked. For example, schema-based planners generate discourse by instantiating the rhetorical predicates of schemata [McKeown, 1985, Paris, 1988, McCoy, 1989]; once the schemata are selected and their predicates are instantiated, the content and organization of the discourse are fixed. More recent, hierarchical discourse planners [Hovy, 1988, Moore and Swartout, 1991] are also monotonic. Given a rhetorical goal, these planners subgoal to construct a tree whose leaves are specifications for producing small pieces of text; they never revoke their selection of operators or change their binding of operator variables.

Because these approaches to discourse planning make no provision for evaluating and modifying a plan once it has been constructed, their flexibility is limited. Therefore, they rely on designers to develop schemata and planning operators that always "get it right" the first time. As the length of the text to be generated expands, it becomes increasingly difficult to anticipate all planning contingencies. This is especially true for a discourse planner that is supposed to be *context-sensitive*, e.g., the discourse planner for an ITS that customizes the content and organization of its explanations for a student's current mastery of domain concepts.

What's needed is a more fluid method of discourse production, such as the one employed by human writers. Researchers who study the writing process argue that revision plays a crucial role in composition [Hayes

---

*This research is supported by the National Science Foundation under grant IRI-8620052; the Air Force Human Resources Laboratory under grant F33615-90-C-0014; and donations from the research laboratories of Hughes and GTE.

[1] We follow the typical decomposition of natural language generation into *planning*, which is concerned with content determination and high-level organization issues, and *realization*, which is concerned with sentential organization, pronominalization, ellipsis, and lexical choice.

and Flower, 1986]. Although a good case has been made that natural language generators can benefit substantially from revision [Yazdani, 1987], only a few projects have explored revision computationally [Mann and Moore, 1981, Vaughan and McDonald, 1986, Wong and Simmons, 1988, Gabriel, 1988]. These projects contribute to our understanding of the revision process, but they focus on local sentential transformations, such as forming complex sentences from simple ones and combining two sentences with the same verb phrases, rather than on methods for globally restructuring discourse.

We have designed a *revision-based* model of discourse planning that constructs multi-paragraph discourse plans, evaluates them, and globally restructures them. The planner can revise its plans because it delays organizational commitments as long as possible and interleaves the content determination and organization activities. The model has been implemented in an experimental system, KNIGHT[2], a context-sensitive discourse generator for ITSs. KNIGHT generates multi-paragraph explanations in the domain of biology.

Revision-Based Production

The *revision-based* planning algorithm controls the assembly of discourse plans. To provide organizational flexibility, it delays organizational commitments as long as possible. By delaying organizational commitments, it can more easily revise plans to improve their organization and to include additional content. Initially, the *discourse-elements*—specifications for sentences to be generated—of an evolving plan are placed in the *planning space* in a loosely organized fashion (Figure 1). For organizational purposes, the planning space is partitioned into *clusters*, each of which will eventually result in a paragraph of text. As the plan develops, the planner adds new elements to the plan, transfers elements from one cluster to another, and creates new clusters as needed. Thus it interleaves the construction and organization of elements. It gradually imposes increasing order on the elements until they are sequential, at which time it passes the elements to the *realizer*, which translates them into text.

The planning algorithm is shown in Figure 2. Planning begins when a query is posed. The planner first constructs the initial group of discourse-elements by identifying the relevant knowledge in the Botany Knowledge Base, a large-scale, frame-based representation that we and our colleagues have been constructing for the past four years [Porter *et al.*, 1988]. This knowledge base contains over two thousand frames about botanical objects (i.e., plant anatomy) and the processes that change them (i.e., plant physiology and development). Next, the planner evaluates the initial plan to determine if some elements can be rearranged to improve the organization. If so, it considers forming new clusters and transferring elements among clusters. If the plan was restructured, the planner determines if the reorganized plan should include additional content. If so, the planner applies *elaborators* that construct additional elements and inserts them into the plan. It then re-evaluates the plan to determine if any coherence has been lost. If repairs are called for, it installs the necessary patches, which take the form of outline statements, transition statements, and summaries. Finally, it schedules the elements in each cluster and passes them to the realizer.

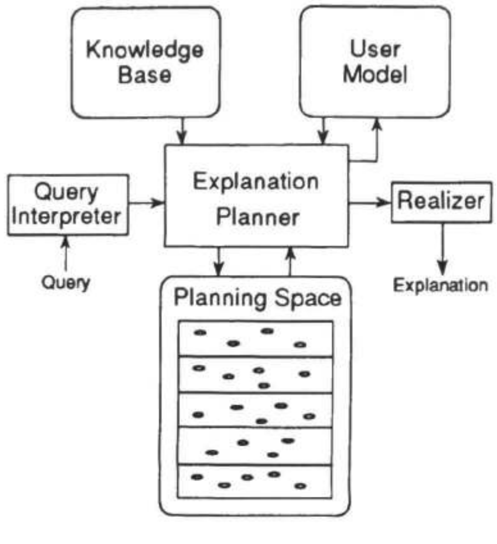

Figure 1: The architecture

```
Plan ← construct-elements(Query);
repeat
        for each restructuring operator R do
            if applicable(R, Plan) then
                Plan ← restructure-plan(Plan, R);
                for each elaborator E relevant to R do
                    if applicable(E, Plan) then
                        Plan ← elaborate-plan(Plan, E);
        if changed Plan then
            for each patching operator P do
                if applicable(P, Plan) then
                    Plan ← patch-plan(Plan, P)
until no more changes
for each Cluster in Plan do
        Cluster ← schedule-elements(Cluster);
        realize(Cluster)
```

Figure 2: The explanation planning algorithm

Planning a Discourse

To demonstrate KNIGHT's operation, it was posed the query, "How does an embryo sac form?" The relevant portion of the knowledge base is shown in Fig-

---

[2] KNowledge-Integration-based Generator of History-sensitive Text

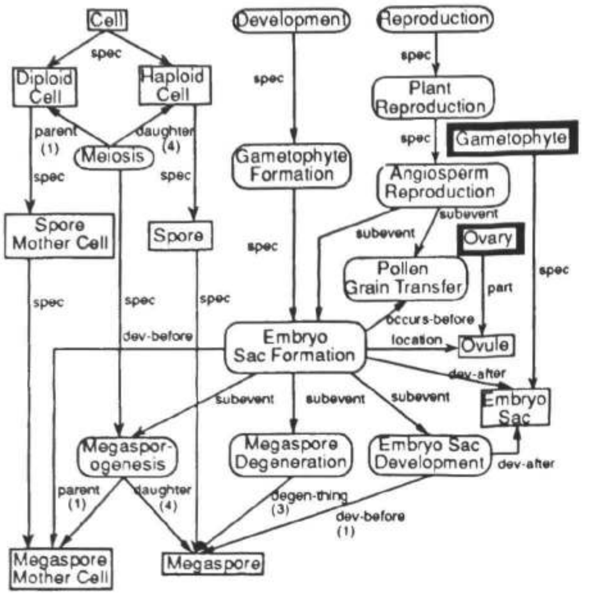

Figure 3: Portion of knowledge base and student model on embryo sac formation

ure 3. (The concepts that were familiar to the student are indicated by the darkened outlines.) The explanation KNIGHT constructed is shown in Figure 4.

Initial Plan Construction

The planner constructs the initial plan by selecting subgraphs of the knowledge base, constructing discourse-elements from these propositions, and inserting the elements into clusters. To determine the initial content of the discourse, KNIGHT uses *graph schemata* to select the core of the explanation. It instantiates the graph schema associated with the type of the query to extract subgraphs of the knowledge base. For example, to explain a process *P*, KNIGHT applies a *process-description* graph schema to define a graph consisting of *P* itself, the actors in *P*, the location of *P*, the subprocesses of *P*, and the subprocesses' actors.[3]

As it selects the relevant subgraphs of the knowledge base, KNIGHT constructs discourse-elements from the subgraphs' constituent propositions, and it assigns them to clusters in the planning space. The types of clusters, as well as their order, depend on the type of query issued. For example, the cluster types associated with the `Describe process` query type are the *Motivation* cluster, which conveys the overall significance of the process; the *Introduction* cluster, which provides an overview of the process; and the *Mechanism* cluster, which discusses the mechanism by which the process occurs. These cluster types were derived from an analysis of discourse patterns found in science textbooks. Figure 5 shows a snapshot of the planning space after the initial construction phase. Monotonic planners terminate after this phase, but KNIGHT evaluates and revises plans to improve them.

STUDENT: How does an embryo sac form?
KNIGHT:

Reproduction is a central function of biological entities. One kind of reproduction is angiosperm sexual reproduction. Embryo sac formation is a step of angiosperm sexual reproduction. Before explaining embryo sac formation, we will first discuss some prerequisite information about cells.

Two kinds of cells are a diploid cell and a haploid cell. A diploid cell has 2 sets of chromosomes in its nucleus, but a haploid cell has 1 set of chromosomes in its nucleus. During meiosis, a diploid cell reproduces, resulting in 4 haploid cells. One kind of diploid cell is a megaspore mother cell, which is involved in embryo sac formation. One kind of haploid cell is a megaspore, which is involved in embryo sac formation.

Now we can explain embryo sac formation. Embryo sac formation is a kind of development. It takes place in an ovule. An ovule is part of an ovary. During embryo sac formation, a megaspore mother cell develops into an embryo sac, which is a gametophyte.

Embryo sac formation consists of megasporogenesis, megaspore degeneration, and embryo sac development. First, during megasporogenesis, a megaspore mother cell reproduces, resulting in 4 megaspores. Later, during megaspore degeneration, 3 megaspores degenerate. During embryo sac development, 1 megaspore develops into an embryo sac.

Let's review the key points. Reproduction is a central function of biological entities. One kind of reproduction is angiosperm sexual reproduction. Embryo sac formation is a step of angiosperm sexual reproduction. Embryo sac formation is a kind of development. During embryo sac formation, a megaspore mother cell develops into an embryo sac.

Figure 4: Explanation of embryo sac formation

Restructuring Plans

Once the planner has completed the initial construction phase, it evaluates the plan to determine if it can make improvements. It decides if a revision should be made by determining the conceptual similarity of discourse-elements in different clusters. If elements from different clusters satisfy the similarity criteria, it forms a new cluster by grouping the similar elements and positioning the new cluster in an appropriate location in the plan.

KNIGHT's criteria for conceptual similarity are based on our theory of *view types* [Souther *et al.*, 1989], which gauge the coherence of groups of concepts. Each view type aggregates concepts according to a particu-

---

[3] Because KNIGHT is a context-sensitive planner, it uses specialized selection strategies to determine additional subgraphs that are appropriate for the student. We discuss these methods for context-sensitive discourse planning elsewhere [Lester and Porter, 1991].

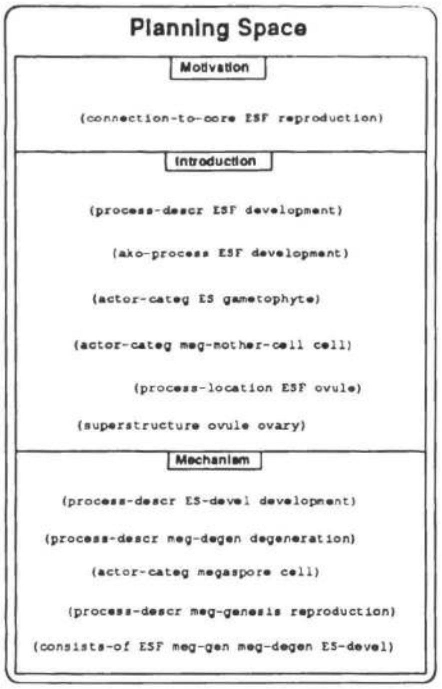

Figure 5: Plan after initial construction phase

lar perspective. For example, the *categorical* view type aggregates a concept and the concepts in its taxonomic heritage. Hence, KNIGHT uses a restructuring operator to group "categorically" similar information. Categorical restructuring is accomplished by finding two discourse-elements, each of which discusses two concepts as being specializations of a common concept, and grouping them together. In attempting to apply the categorical restructuring operator, KNIGHT's search through the planning space reveals that a categorical element in the *Introduction* cluster explains that a *megaspore mother cell* is a kind of *cell*, and another categorical element in the *Mechanism* cluster explains that a *megaspore* is a kind of *cell*. Rather than distributing the discussion across two possibly distant paragraphs, grouping the elements together provides a coherent forum for presenting a top-down taxonomic discussion of cells.

Of course, if the designers of the construction methods could have anticipated this situation, the categorical cluster could have been included in the initial plan. However, the need for such a cluster must be determined at run-time; only then can it be known that there are elements that fortuitously explain two concepts in terms of the same generalization.

Elaborating Plans

In addition to improving the organization of the discourse, restructuring also provides opportunities for including additional content. This is particularly significant for discourse planners used by ITSs because it allows them to interject important supplementary information that is important in the domain. The planner uses *elaborators* for this purpose.

For example, after a categorical restructuring has been performed, as described in the previous section, the plan will include a cluster that discusses the subclasses of a particular concept. An appropriate elaboration might add discourse-elements that compare the subclasses. KNIGHT uses such a *comparison* elaborator. Given two categorical discourse-elements that explain two concepts in terms of the same generalization, KNIGHT searches for the crucial differences between the two concepts. When it applies the comparison elaborator to the discourse-elements of the categorical cluster, it determines that *megaspore mother cell* and *megaspore* are both kinds of a *cell*, and that these concepts begin to differ in the taxonomy at the concepts *diploid cell* and *haploid cell*. It constructs the discourse-elements that discuss the two kinds of cells and their differences and adds these elements to the newly created *Categorical* cluster.

Finalizing Plans

The ability to dynamically restructure discourse plans and to generate elaborations on the fly provides high flexibility. However, flexibility is sometimes achieved at the expense of coherence. In these cases, the planner applies *patching* operators to restore coherence. For example, it constructs an outline element and adds it to the *Motivation* cluster, it constructs a transition element and adds it to the *Introduction* cluster, and it constructs a summary cluster and attaches it to the end of the plan. The final plan is shown in Figure 6.

The final step is to linearize the elements within a cluster. The planner invokes its *scheduling* operators to provide this service. Each scheduling operator is based on a commonly occurring pattern of text. For example, textbooks typically discuss the subprocesses of a process in the order in which they occur, so KNIGHT's *Mechanism* scheduling operator imposes a temporal order on the elements that discuss the subprocesses of a process.

Conclusion

ITSs in scientific domains require explanation generators that can produce coherent multi-paragraph explanations. The revision-based model of discourse planning satisfies this requirement. KNIGHT's output was favorably evaluated by a domain expert, who analyzed three explanations it generated, including the one discussed here. He found the explanations to be both accurate—their statements were scientifically correct—and clear—their presentations were easily understandable.

The planner achieves these results by dynamically restructuring discourse plans. Restructuring is possible because it delays organizational commitments. In

Figure 6: The final plan

contrast to monotonic discourse planners, the revision-based model allows decisions about organization to be evaluated and revised at run-time. This accords well with research on writing. In short, revision-based planning offers cognitive plausibility and constitutes a promising approach to multi-paragraph discourse generation.

Acknowledgements

We are grateful to those responsible for the construction of the Botany Knowledge Base, including Art Souther, Karen Pittman, Ken Murray, Liane Acker, Brad Blumenthal, Dave Severinson, and Tom Jones. KNIGHT was implemented using Theo, a knowledge base access tool provided by Tom Mitchell. Thanks to Erik Eilerts for extending Theo's capabilities, and to Brad Blumenthal for his Lisp wizardry. Thanks also to Rich Mallory, Kathy Mitchell, Ken Murray, Hwee Tou Ng, Jeff Rickel, Wing Wong, and especially to Liane Acker for helpful discussions and comments on earlier drafts of this paper.

References

[Gabriel, 1988] Richard P. Gabriel. Deliberate writing. In David D. McDonald and Leonard Bolc, editors, *Natural Language Generation Systems*, pages 1–46. Springer-Verlag, New York, 1988.

[Hayes and Flower, 1986] John R. Hayes and Linda S. Flower. Writing research and the writer. *American Psychologist*, 41:1106–1113, 1986.

[Hovy, 1988] Eduard H. Hovy. Planning coherent multisentential text. In *Proceedings of the 26th Annual Meeting*, pages 163–169, Buffalo, 1988. Association for Computational Linguistics.

[Lester and Porter, 1991] James C. Lester and Bruce W. Porter. A student-sensitive discourse generator for intelligent tutoring systems. In *Proceedings of the International Conference on the Learning Sciences*, 1991.

[Mann and Moore, 1981] William C. Mann and James A. Moore. Computer generation of multiparagraph english text. *American Journal of Computational Linguistics*, 7(1):17–29, 1981.

[McCoy, 1989] Kathleen F. McCoy. Generating context-sensitive responses to object-related misconceptions. *Artificial Intelligence*, 41:157–195, 1989.

[McKeown, 1985] Kathleen R. McKeown. *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text.* Cambridge University Press, 1985.

[Moore and Swartout, 1991]
Johanna D. Moore and William R. Swartout. A reactive approach to explanation: Taking the user's feedback into account. In Cécile L. Paris, William R. Swartout, and William C. Mann, editors, *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, pages 3–48. Kluwer Academic Publishers, Boston, 1991.

[Paris, 1988] Cécile L. Paris. Tailoring object descriptions to a user's level of expertise. *Computational Linguistics*, 14(3):64–78, September 1988.

[Porter *et al.*, 1988] Bruce Porter, James Lester, Kenneth Murray, Karen Pittman, Art Souther, Liane Acker, and Tom Jones. AI research in the context of a multifunctional knowledge base: The botany knowledge base project. Technical Report AI Laboratory AI88-88, University of Texas at Austin, Austin, Texas, 1988.

[Souther *et al.*, 1989] Art Souther, Liane Acker, James Lester, and Bruce Porter. Using view types to generate explanations in intelligent tutoring systems. In *Proceedings of the Eleventh Cognitive Science Society Conference*, pages 123–130, 1989.

[Vaughan and McDonald, 1986] Marie M. Vaughan and David D. McDonald. A model of revision in natural language generation. In *Proceedings of the 24th Annual Meeting*, pages 90–96, Columbia University, 1986. Association for Computational Linguistics.

[Wenger, 1987] Etienne Wenger. *Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge.* Morgan Kaufmann, Los Altos, California, 1987.

[Wong and Simmons, 1988] Wing-Kwong C. Wong and Robert F. Simmons. A blackboard model of text production with revision. In *Proceedings of the AAAI Workshop on Text Planning and Realization*, St. Paul, Minnesota, August 1988.

[Yazdani, 1987] Masoud Yazdani. Reviewing as a component of the text generation process. In Gerard Kempen, editor, *Natural Language Generation*, pages 183–190. Martinus Nijhoff, Dordrecht, The Netherlands, 1987.