Artificial Intelligence and Law **6:** 111–149, 1998.

Automating Judicial Document Drafting: A Discourse-Based Approach

L. KARL BRANTING
Department of Computer Science, University of Wyoming, Box 3682, Laramie, WY 82071,
E-mail: karl@index.uwyo.edu

JAMES C. LESTER and CHARLES B. CALLAWAY
Department of Computer Science, North Carolina State University, Box 8206, Raleigh, NC 27606,
E-mail: lester@adm.csc.ncsu.edu, cbcallaw@eos.ncsu.edu

Abstract. Document drafting is a central judicial problem-solving activity. Development of automated systems to assist judicial document drafting has been impeded by the absence of an explicit model of (1) the connection between the document drafter's goals and the text intended to achieve those goals, and (2) the rhetorical constraints expressing the stylistic and discourse conventions of the document's genre. This paper proposes a model in which the drafter's goals and the stylistic and discourse conventions are represented in a *discourse structure* consisting of a tree of illocutionary and rhetorical operators with document text as leaves. A *document grammar* based on the discourse structures of a representative set of documents can be used to synthesize a wide range of additional documents from sets of case facts. The applicability of this model to a representative class of judicial orders – jurisdictional show-cause orders – is demonstrated by illustrating (1) the analysis of show-cause orders in terms of discourse structures, (2) the derivation of a document grammar from discourse structures of two typical show-cause orders, and (3) the synthesis of a new show-cause order from the document grammar.

1. Introduction

Legal problem solving subsumes a number of distinct tasks, including analyzing the legal consequences of actual or hypothetical sequences of actions, argumentation, advising clients, planning transactions, and drafting legal documents. Legal document drafting is an essential professional skill for attorneys and judges. In the U.S., a significant portion of attorneys' workloads consists of drafting documents intended to precisely stipulate legal relationships such as wills, contracts, and leases, and persuasive documents arising from litigation such as pleadings, motions, and briefs.

Document drafting is a central activity of the judiciary. Judges' resolutions of the disputes that come before them are generally embodied in written documents. These documents can vary in complexity from brief memos to lengthy appellate opinions. While judges have primary responsibility for judicial document drafting, they are assisted by various judicial staff members including administrative and secretarial staff and law clerks.

Two factors impose very high requirements for correctness and consistency on judicial documents. First, the Anglo-American system embraces the doctrine of *stare decisis* under which judicial decisions can be used as an authority to resolve subsequent disputes. As a result, the impact of a document may extend far beyond the parties whose dispute gave rise to the document. Second, all judicial decisions and orders except those of the highest court in a given jurisdiction are subject to review by higher courts. A party adversely affected by a judicial decision has a strong incentive to discover any error or inconsistency in the document embodying the decision, since such error or inconsistency could be used to attack the decision in a higher court. Thus, high standards of correctness and consistency are essential in judicial document drafting.

Document drafting can be viewed as a kind of configuration task in which textual elements are selected and arranged to satisfy the goals of the drafter and to conform to the stylistic conventions of the document genre. One source of complexity in document drafting is the combinatorics of selection and configuration decisions, which create large search spaces characteristic of most synthesis tasks. However, a more fundamental reason for the difficulty of document drafting is that the goals that documents are intended to achieve and the stylistic conventions to which they must conform are seldom made explicit. An explicit representation of these goals and conventions is essential to the development of automated tools to assist in the document drafting process.

This paper proposes a model of documents that makes the underlying goals and conventions explicit and uses this explicit theory to assist in the construction of new documents. These goals and conventions are expressed as a dependency tree having two types of discourse operators as interior nodes: *illocutionary* operators, which express the goals that a document achieves; and *rhetorical* operators, which express the stylistic conventions of the document's genre[1] We term this dependency tree the *discourse structure* of the document. The illocutionary and rhetorical operators capable of describing the discourse structures of a set of documents constitutes a *document grammar* for the documents.

We have previously argued that representation of the discourse structure of documents can facilitate (1) retrieval, interpretation, and adaptation of previous documents, (2) maintenance of multi-generation documents, and (3) comparison of documents at a deeper level than mere surface text (Branting and Lester, 1996a; Branting and Lester, 1996b). In this paper we focus on the task of drafting new documents using a document grammar derived from the discourse structures of a set of documents representative of a given document genre.

[1] An illocutionary operator is a speech act such as informing, requesting, warning, or promising. A rhetorical operator is a discourse or coherence relation, such as exemplification, generalization, sequence, or elaboration. See (Allen, 1987) for a more detailed discussion.

Section 2 describes a representative class of judicial documents – appellate jurisdictional show-cause orders – and describes the potential benefits of automating their creation. Section 3 illustrates how the show-cause orders can be represented in terms of discourse structures and describes informally how a document grammar derived from these discourse structures can be used in the drafting of new show-cause orders. Section 4 describes our implementation of a prototype unification-based document planning system, the DOCU-PLANNER, and presents details of the process whereby a document grammar can be used to draft a wide range of related documents. Related work is discussed in Section 5, and Section 6 sets forth the scope of this approach and future research.

2. Judicial Document Drafting

Judicial decisions are expressed in judicial documents. In the Anglo-American legal system, the most prominent judicial documents are appellate decisions. Appellate opinions typically contain a summary of the facts of the case, identification of the issues of law raised in arguments by counsel for each of the parties, pronouncement of the legal propositions supported by the controlling authorities, and declaration of a decision that resolves the issues by applying the legal propositions to the facts of the case (Branting, 1993b). The complexity and individuality of appellate opinions makes automated assistance for such documents far beyond the scope of current technology. However, courts produce a number of other more routine documents having considerable stylistic and substantive consistency, including various types of orders issued in response to motions or *sua sponte*. A single case may give rise to numerous motions at both the trial and appellate levels relating to, e.g., pleadings, discovery, time extensions, motions for dismissal or summary judgment, or sanctions for violations of trial or appellate rules. The rulings of trial and appellate courts on these motions typically take the form of orders of varying degrees of length and complexity. Jurisdictional show-cause orders are typical of such orders.

Jurisdictional show-cause orders are generally issued during jurisdictional screening, a process of determining whether the requirements for an appeal have been satisfied. Jurisdictional screening is typically performed at the earliest possible stage of an appeal to permit cases with jurisdictional defects to be recognized as soon as possible. This minimizes unnecessary consumption of limited judicial resources.

In this paper, discussion of appellate jurisdictional screening will focus on the Colorado Court of Appeals, where one of the authors, Karl Branting, worked for several years as a staff attorney. The Colorado Court of Appeals typically receives over 100 new cases per month. Screening these appeals is too complex for clerical personnel, but must instead be performed by a staff attorney. The staff attorney examines the case file to determine whether the subject matter, finality, and timeliness requirements for appellate jurisdiction have been met. If there appears to be a jurisdictional defect, the staff attorney drafts a show-cause order that sets forth the apparent defect and orders the appellant to rebut the defect within a fixed time period or face dismissal of the appeal.

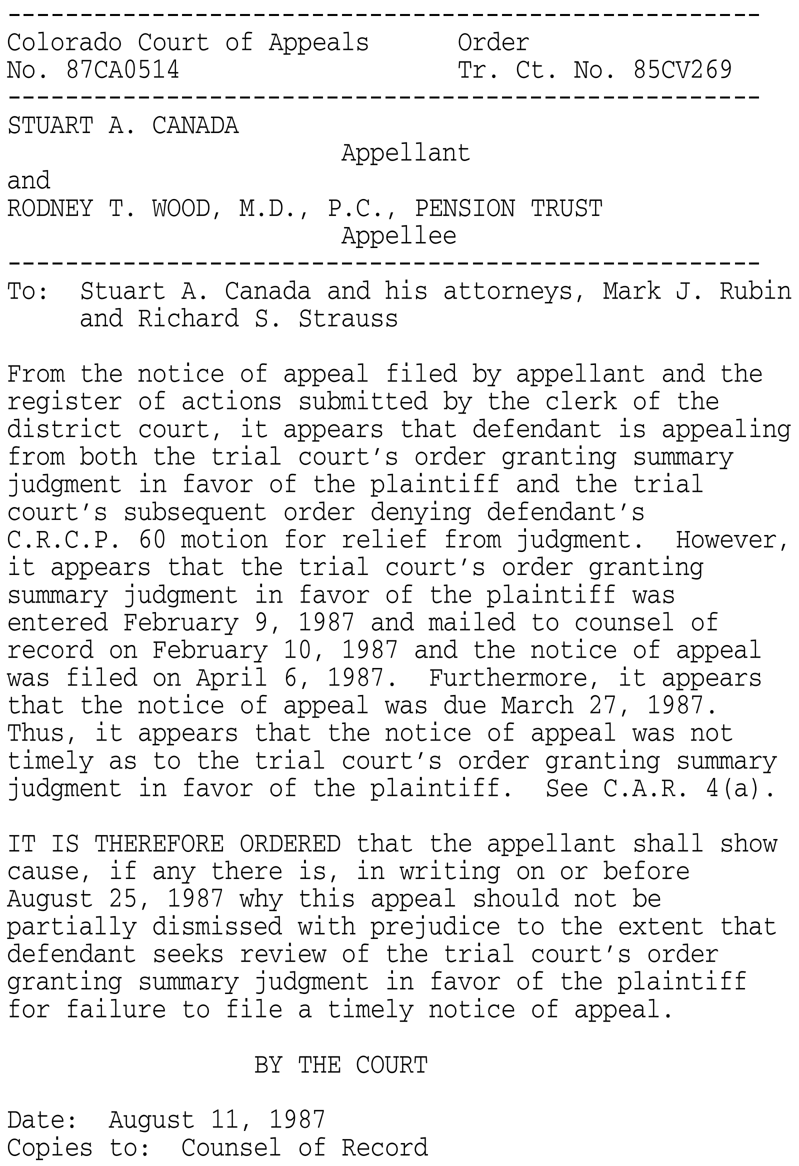
Figure 1 shows a typical show-cause order. This order identifies an apparent defect – an untimely notice of appeal – and orders the appellant to show cause, within 14 days, why the appeal should not therefore be dismissed. Show-cause orders typify legal documents that are produced in relatively high volume (several hundred per year), are complex enough to require drafting by an attorney, yet have a high degree of stylistic and substantive consistency.

Currently, staff attorneys at the Colorado Court of Appeals draft show-cause orders manually. One approach is for the staff attorney to reuse only those "boilerplate" text blocks that are common to all show-cause orders (e.g., "From the notice of appeal filed by appellant ... "). Alternatively, a staff attorney may refer to a collection of previous show-cause orders. The attorney can search this collection for a previous order that involved jurisdictional defects similar to those in the current case. The reusable language from the previous case may then be transcribed into the current case, or the staff attorney may photocopy the previous order, cross out the portions inapplicable to the new case, and write in portions specific to the new case.

Manual drafting of show-cause orders has several clear disadvantages. First, the process is very time-consuming and laborious. The first approach, which reuses only the most general boilerplate language, entails repeated replication of drafting effort and creates a high likelihood of inconsistent language. The second approach, reuse of similar orders, depends on the staff attorney's ability to find and appropriately modify previous orders. This in turn depends on the attorney's ability to understand the relevant similarities and differences between the goals that must be achieved by the current show-cause order and the intentions underlying previous orders.

The difficulty of drafting correct and consistent documents is exacerbated by frequent personnel changes. For example, at the Colorado Court of Appeals, few staff attorneys are willing to do jurisdictional screening for more than six months, and many do screening for as little as three months. As a result, jurisdictional screening is typically performed by attorneys with only limited experience drafting show-cause orders.

Drafting even routine judicial documents, such as show-cause orders, is laborious and error-prone for an inexperienced drafter. A high volume of routine orders may therefore constitute a significant drain on judicial resources even if each individual order is relatively straightforward. It is widely recognized that rising caseloads constitute one of the most pervasive problems confronting the judicial system in the United States (Snellenburg, 1989). Technologies to enable judges to use their time and expertise as efficiently as possible are therefore of great potential importance to the judiciary. Automating the drafting of relatively routine documents could make a significant contribution to judicial efficiency.



```
----------------------------------------------------
Colorado Court of Appeals      Order
No. 87CA0514                   Tr. Ct. No. 85CV269
----------------------------------------------------
STUART A. CANADA
                       Appellant
and
RODNEY T. WOOD, M.D., P.C., PENSION TRUST
                       Appellee
----------------------------------------------------
To:  Stuart A. Canada and his attorneys, Mark J. Rubin
     and Richard S. Strauss

From the notice of appeal filed by appellant and the
register of actions submitted by the clerk of the
district court, it appears that defendant is appealing
from both the trial court's order granting summary
judgment in favor of the plaintiff and the trial
court's subsequent order denying defendant's
C.R.C.P. 60 motion for relief from judgment.  However,
it appears that the trial court's order granting
summary judgment in favor of the plaintiff was
entered February 9, 1987 and mailed to counsel of
record on February 10, 1987 and the notice of appeal
was filed on April 6, 1987.  Furthermore, it appears
that the notice of appeal was due March 27, 1987.
Thus, it appears that the notice of appeal was not
timely as to the trial court's order granting summary
judgment in favor of the plaintiff.  See C.A.R. 4(a).

IT IS THEREFORE ORDERED that the appellant shall show
cause, if any there is, in writing on or before
August 25, 1987 why this appeal should not be
partially dismissed with prejudice to the extent that
defendant seeks review of the trial court's order
granting summary judgment in favor of the plaintiff
for failure to file a timely notice of appeal.

                 BY THE COURT

Date:  August 11, 1987
Copies to:  Counsel of Record
```

Figure 1. A typical show-cause order.

One approach to providing automated assistance in the drafting of show-cause orders would be to design a set of templates for various show-cause orders, e.g., using WordPerfect macros. However, the wide variety of possible jurisdictional defects and the even wider variety of factual situations that can give rise to jurisdictional defects make devising an adequate collection of macros impracticable. Moreover, even if a sufficient set of macros could be created, these macros would present the user with an overwhelming number of choices, making selection of the most appropriate macro unlikely.

We argue that the goals of accuracy, efficiency, and stylistic consistency are best served by an approach to drafting routine legal documents that is based on the discourse structure described in the next section. Our model of the document drafting task is as follows:

Given:

- A set of relevant case facts.
- One or more illocutionary goals that the document is to achieve, e.g., establishing the prerequisites for dismissal.
- A document grammar which includes:
  - illocutionary operators embodying the substantive legal rules governing documents in the genre, and
  - rhetorical operators embodying the stylistic and discourse conventions of the genre.

Do:

- Planning. Find and instantiate a set of illocutionary operators that achieves the document's illocutionary goals in terms of the given case facts and the applicable set of rhetorical operators.
- Drafting. Synthesize a document that satisfies the illocutionary and rhetorical operators.

There are several possible sources of the relevant case facts. One approach is a conventional backward-chaining rule-based system, which would chain through the applicable jurisdictional rules querying the user when necessary. Interactions under this approach would be system-initiated. JEDA (Pethe et al., 1989) and LAW CLERK (Branting, 1993a) illustrate system-initiated acquisition of case facts for document drafting. An alternative, user-initiated, approach would use an electronic form with entries corresponding to possible case values. A third alternative is a mixed-initiative approach under which the user can directly provide case facts, ask the system for examples, counter-examples and advice on answering questions, or invoke an inference engine to help infer facts.[3] The discussion below assumes only that the relevant case facts have been obtained through one of these mechanisms, but does not depend on the particular mechanism used.

The next section illustrates informally how the illocutionary and rhetorical goals of show-cause orders can be represented by a discourse structure and how a document grammar derived from this structure can be used to draft new documents. A more detailed description of this process is set forth in Section 4, and a complete document grammar for the examples in this paper appears in Appendix A.

[3] This approach is used in the Jurisdictional Screening Assistant (JSA), a decision support system for appellate jurisdiction screening under development at the University of Wyoming Department of Computer Science.

3. Document Drafting Using Discourse Structures

3.1. ILLOCUTIONARY AND RHETORICAL STRUCTURE OF PERFORMATIVE DOCUMENTS

Legal documents can serve a variety of illocutionary goals, including eliciting information, persuading, memorializing events such as reciprocal communications, or accomplishing performative goals, such as creating or revoking legal relationships. Judicial orders typically have a performative objective: they are intended to define or alter legal relationships relevant to some controversy before the court.

There are generally three requirements that a performative judicial document, such as an order or decision, must have to achieve the goal of defining or altering a legal relationship. First, the document must find that some set of relevant facts is present in the case. Second, the document must rule that one or more legal propositions follow from applicable legal warrants under these facts. Finally, the document must order some legal consequence justified by the legal propositions under the given facts.[4]

For example, the show-cause order set forth in Figure 2 finds that the summary judgment from which Appellant is appealing was granted on February 9, 1987 and mailed to Appellant on February 10, 1987, and that Appellant's notice of appeal was filed on April 6, 1987. The show-cause order rules that the notice of appeal was due on March 27, 1987 (45 days after notification of the judgment by mailing). Finally, the Appellant is ordered to show why the findings or rulings are not justified or suffer the sanction of dismissal with prejudice of the portion of the appeal for which the notice of appeal was untimely.

The illocutionary goal of the show-cause order as a whole is to establish the prerequisites for dismissal of the appeal. The findings and rulings required to establish the prerequisites for dismissal are determined by the legal rules governing jurisdiction in the Court of Appeals. These are set forth in the Colorado Rules of Civil Procedure (C.R.C.P) and the Colorado Appellate Rules (C.A.R.). For example, C.A.R. 4(a) provides that the notice of appeal "shall be filed with the appellate court . . . within forty-five days." The commencement of the 45 day period is triggered by (1) "entry of the judgment or order appealed from" if the parties are present at the time the judgment or order is announced, or (2) "the date of the mailing of the notice" of judgment, if the notice is transmitted to the parties by mail.

The illocutionary structure of the *Canada v. Wood* show-cause order is shown on the left side of Figure 2. The top-level illocutionary goal is to establish

[4] The findings of facts are analogous to the data in Toulmin's (Toulmin, 1958) model of argument. The rulings of law are analogous to Toulmin's warrant and claim, since the rulings comprise both the applicable legal authority and the conclusion that follows from applying the authority to the facts. The additional element of performative judicial documents, the order, arises from courts' institutional power to actually bring about changes in legal relations through documents of an appropriate structure. The order is the final element necessary to bring about such a change.

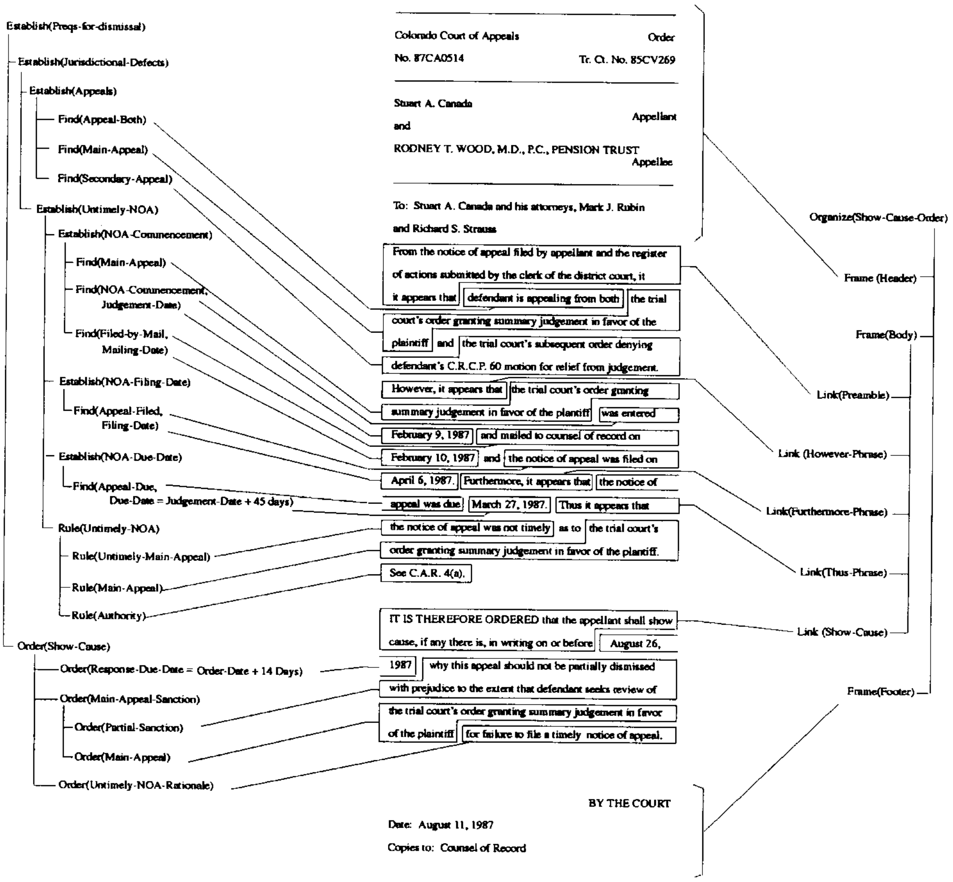

Establish(Preqs-for-dismissal)
Establish(Jurisdictional-Defects)
Establish(Appeals)
Find(Appeal-Both)
Find(Main-Appeal)
Find(Secondary-Appeal)
Establish(Untimely-NOA)
Establish(NOA-Commencement)
Find(Main-Appeal)
Find(NOA-Commencement, Judgement-Date)
Find(Filed-by-Mail, Mailing-Date)
Establish(NOA-Filing-Date)
Find(Appeal-Filed, Filing-Date)
Establish(NOA-Due-Date)
Find(Appeal-Due, Due-Date = Judgement-Date + 45 days)
Rule(Untimely-NOA)
Rule(Untimely-Main-Appeal)
Rule(Main-Appeal)
Rule(Authority)
Order(Show-Cause)
Order(Response-Due-Date = Order-Date + 14 Days)
Order(Main-Appeal-Sanction)
Order(Partial-Sanction)
Order(Main-Appeal)
Order(Untimely-NOA-Rationale)
Colorado Court of Appeals
Order
No. 87CA0514
Tr. Ct. No. 85CV269
Stuart A. Canada
Appellant
and
RODNEY T. WOOD, M.D., P.C., PENSION TRUST
Appellee
To: Stuart A. Canada and his attorneys, Mark J. Rubin
and Richard S. Strauss
From the notice of appeal filed by appellant and the register
of actions submitted by the clerk of the district court, it
it appears that
defendant is appealing from both
the trial
court's order granting summary judgement in favor of the
plaintiff
and
the trial court's subsequent order denying
defendant's C.R.C.P. 60 motion for relief from judgement.
However, it appears that
the trial court's order granting
summary judgement in favor of the plantiff
was entered
February 9, 1987
and mailed to counsel of record on
February 10, 1987
and
the notice of appeal was filed on
April 6, 1987.
Furthermore, it appears that
the notice of
appeal was due
March 27, 1987.
Thus it appears that
the notice of appeal was not timely
as to
the trial court's
order granting summary judgement in favor of the plantiff.
See C.A.R. 4(a).
IT IS THEREFORE ORDERED that the appellant shall show
cause, if any there is, in writing on or before
August 26,
1987
why this appeal should not be partially dismissed
with prejudice to the extent that defendant seeks review of
the trial court's order granting summary judgement in favor
of the plaintiff
for failure to file a timely notice of appeal.
BY THE COURT
Date: August 11, 1987
Copies to: Counsel of Record
Organize(Show-Cause-Order)
Frame (Header)
Frame(Body)
Link(Preamble)
Link (However-Phrase)
Link(Furthermore-Phrase)
Link(Thus-Phrase)
Link (Show-Cause)
Frame(Footer)

Figure 2. The illocutionary and rhetorical structure of *Canada*

the prerequisites for dismissal. This goal has two subgoals: to establish the existence of a jurisdictional defect, and to order an appropriate sanction. The relationship between an illocutionary goal, such as Establish(Untimely-notice-of-appeal), and its subgoals, Establish(Notice-of-appeal-commencement-date), Establish(Notice-of-appeal-filing-date), Establish(Notice-of-appeal-due-date), and Rule(Untimely-notice-of-appeal), is expressed in an *illocutionary operator*.

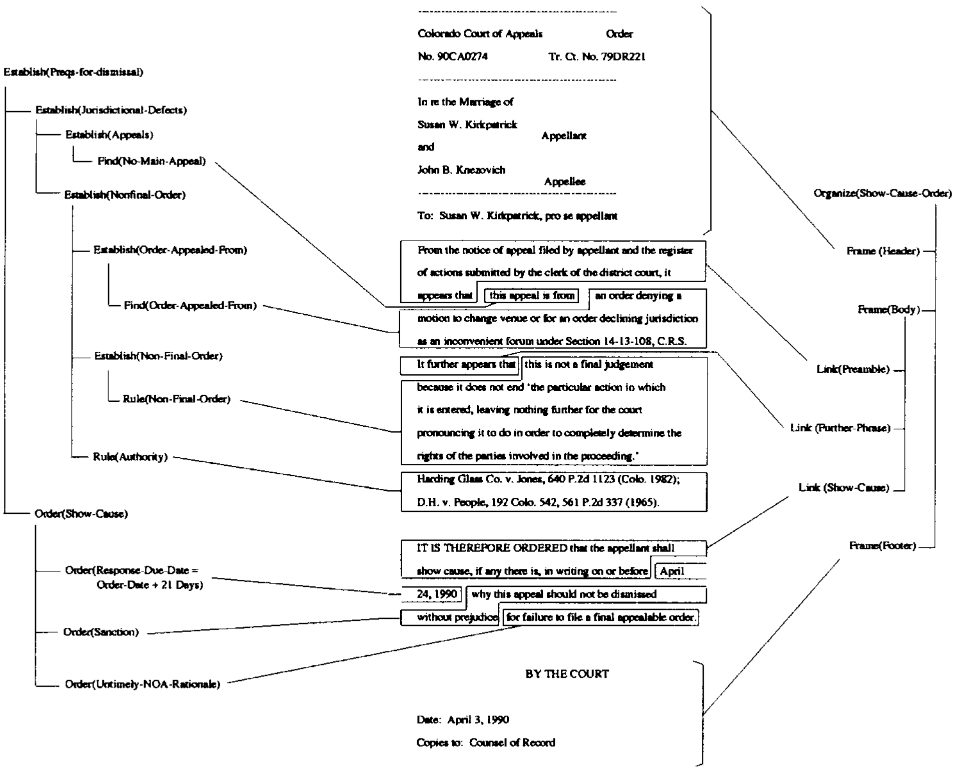
The subtree underneath Establish(Jurisdictional-Defects) is similar to the goal tree that would be generated by a rule-based system for determining the presence of jurisdictional defects. In a conventional goal tree, legal rules would be used to repeatedly decompose a goal establishing a jurisdictional defect into subgoals. Ultimately, these subgoals would be grounded in the facts of the case. An illocutionary structure differs from a conventional goal tree in that the leaf nodes are not limited to case facts, but also include textual elements that satisfy illocutionary goals. For example, under C.A.R. 4(a), determining that a notice of appeal is untimely requires determining the date when the notice of appeal was filed. However, for the document to achieve the illocutionary goal of establishing the date when the notice of appeal was filed, the document must include text that makes a finding of the filing date.

As shown in the left side of Figure 2, the illocutionary goal of establishing a jurisdictional defect has as its subgoal establishing the orders being appealed and establishing that the notice of appeal was untimely as to one of the orders. Establishing untimeliness, in turn, has as subgoals: establishing the commencement date of the time for filing a notice of appeal, establishing the due date of the notice of appeal (45 days after commencement), establishing the actual filing date, and ruling that the actual filing date was after the due date. The subgoals of these goals, in turn, include: finding the judgment, mailing, and filing dates, ruling when the date was due, and ruling that since the filing date was after the due date the notice of appeal was untimely. The second subgoal for establishing the prerequisites for dismissal is the show-cause order, which has as subgoals: ordering a time limit for response, a sanction, and a rationale for the sanction.

In summary, the illocutionary goal structure expresses two kinds of information essential to understanding the structure of a performative judicial document: the goal dependencies among the applicable legal predicates (e.g., timeliness, method of notification of judgment, and commencement of the time for filing a notice of appeal); and the connection between performative text segments and the illocutionary goals that they achieve.

Although the illocutionary goal structure represents information essential to understanding the structure of performative judicial documents such as show-cause orders, this structure is not *per se* sufficient to completely determine the document's surface text. In general, the illocutionary goal structure does not specify rhetorical features such as (1) the order of the textual elements that satisfy various illocutionary goals, and (2) textual elements and stylistic constraints imposed by the particular genre of the text, such as connective phrases and other discourse cues. The right side of Figure 2 shows the rhetorical structure of *Canada v. Wood*. The top-level goal is to Organize(Show-Cause-Order). The subgoals are to provide frames for the caption (header), body, and footer of the order. Within the body, the rhetorical structure includes discourse link features characteristic of the show-cause order genre. Unlike the illocutionary structure, the rhetorical structure is closely connected to the surface text of the document. The relationship between rhetorical goals and their subgoals are expressed by *rhetorical operators*. For example, the top-level rhetorical operator in *Canada* permits the rhetorical goal Organize(show-cause-order) to be reduced to the goals Frame(Header), Frame(body) and Frame(Footer). Together, the illocutionary and rhetorical structures constitute the *discourse structure* of a document.[5]

[5] The discourse structure consists of a single dependency tree with both illocutionary and rhetorical operators as interior nodes. For clarity, the illocutionary and rhetorical operators appearing in the discourse structure are depicted separately in the left and right sides of Figure 2, respectively. Discourse structure is discussed in more detail in Section 4.



Colorado Court of Appeals Order
No. 90CA0274 Tr. Ct. No. 79DR221

In re the Marriage of
Susan W. Kirkpatrick Appellant
and
John B. Knezovich Appellee

To: Susan W. Kirkpatrick, pro se appellant

From the notice of appeal filed by appellant and the register of actions submitted by the clerk of the district court, it appears that this appeal is from an order denying a motion to change venue or for an order declining jurisdiction as an inconvenient forum under Section 14-13-108, C.R.S. It further appears that this is not a final judgement because it does not end 'the particular action in which it is entered, leaving nothing further for the court pronouncing it to do in order to completely determine the rights of the parties involved in the proceeding.' Harding Glass Co. v. Jones, 640 P.2d 1123 (Colo. 1982); D.H. v. People, 192 Colo. 542, 561 P.2d 337 (1965).

IT IS THEREPORE ORDERED that the appellant shall show cause, if any there is, in writing on or before April 24, 1990 why this appeal should not be dismissed without prejudice for failure to file a final appealable order.

BY THE COURT

Date: April 3, 1990
Copies to: Counsel of Record

Establish(Preqs-for-dismissal)
Establish(Jurisdictional-Defects)
Establish(Appeals)
Find(No-Main-Appeal)
Establish(Nonfinal-Order)
Establish(Order-Appealed-From)
Find(Order-Appealed-From)
Establish(Non-Final-Order)
Rule(Non-Final-Order)
Rule(Authority)
Order(Show-Cause)
Order(Response-Due-Date = Order-Date + 21 Days)
Order(Sanction)
Order(Untimely-NOA-Rationale)
Organize(Show-Cause-Order)
Frame (Header)
Frame(Body)
Link(Preamble)
Link (Further-Phrase)
Link (Show-Cause)
Frame(Footer)

Figure 3. The illocutionary and rhetorical structure of *Kirkpatrick*

In summary, the discourse structure of a document consists of illocutionary and rhetorical operators that represent, respectively, (1) the connection between the document drafter's goals and the text intended to achieve those goals, and (2) the rhetorical constraints expressing the stylistic and discourse conventions of the document's genre. The discourse structure grounds out in the text of the document.

3.2. USING DOCUMENT GRAMMARS FOR DRAFTING

The illocutionary and rhetorical operators necessary to construct the discourse structures of a set of documents together constitute a *document grammar* for those documents. To the extent that the document grammar is based on a representative sample of the population of possible documents within the genre, the grammar will be capable of generating a wide range of additional documents as well.

To illustrate this process informally, suppose that a document grammar has been formalized to express the discourse structures of *Canada* and *In re the Marriage of Kirkpatrick,* shown in Figure 3. The process of forming a document grammar entails:

- Analyzing a representative set of documents to determine their illocutionary and rhetorical structures,
- Extracting the illocutionary and rhetorical operators that appear in each document's justification structure, and
- Generalizing the illocutionary and rhetorical operators.

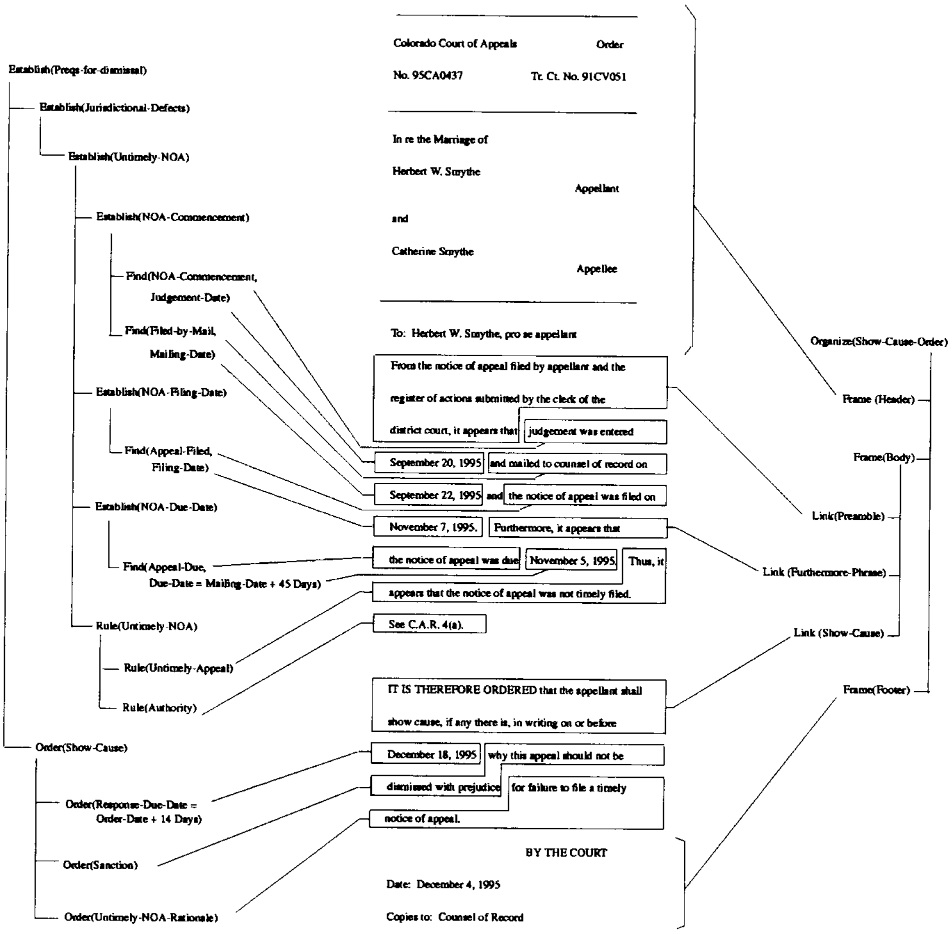
Kirkpatrick's illocutionary structure differs from that of *Canada* in several ways. First, the jurisdictional defect is a lack of finality rather than an untimely notice of appeal. Second, a case that is not yet final may become final at some later time, so the sanction for lack of finality is dismissal *without* prejudice, meaning that the appeal can be filed again at some later date. This is in contrast to an untimely notice of appeal, which can never become timely at a later date and for which the appropriate sanction is therefore dismissal *with* prejudice. Finally, only one order is being appealed in *Kirkpatrick*, so the sanction is not *partial* dismissal, as in *Canada*, but *complete* dismissal.

Kirkpatrick also differs from *Canada* in its rhetorical structure. *Kirkpatrick* is a domestic (i.e., divorce) case and therefore has a different caption than *Canada*, a civil case. Moreover, *Kirkpatrick*'s simpler illocutionary structure results in fewer findings and rulings, so fewer link operators are required.

Suppose that a staff attorney is presented with a file for *In re the Marriage of Herbert W. Smythe and Catherine Smythe*, a domestic case involving a summary judgment entered on September 20, 1995 and mailed to the parties on September 22. Suppose that the appellant, Herbert Smythe, filed a notice of appeal on November 7, 1995, more than 45 days after commencement of the time period for filing a notice of appeal. As discussed above, these facts might be gathered through a decision-support system such as the Jurisdiction Screening Assistant or through some alternative mechanism. Suppose that these facts, together with the docket numbers on appeal and at trial, attorneys' names, *etc.*, are provided to a document drafting system. How could a document grammar for *Kirkpatrick* and *Canada* be applied to these facts to draft an appropriate show-cause order?

The first step is to use the illocutionary operators to create a justification for the goal Establish(Preqs-for-dismissal). This justification, shown on the left side of Figure 4, is similar to the illocutionary structure in *Canada* in that for both cases the jurisdictional defect is an untimely notice of appeal. The structure in *Smythe* is simpler, however, because only a single order is being appealed. Moreover, the existence of only a single appealed order means that sanction in *Smythe* is complete dismissal, as in *Kirkpatrick*, rather than partial dismissal, as in *Canada*. Accordingly, the illocutionary structure combines elements from both *Kirkpatrick* and *Canada*.

The rhetorical structure of *Smythe*, shown on the right side of Figure 4, closely resembles that of *Kirkpatrick* because both are domestic cases involving appeal of a single order. *Smythe*'s illocutionary and rhetorical structures are together sufficient to determine the surface text of the order shown in Figure 4.


Establish(Preqs-for-dismissal)
Establish(Jurisdictional-Defects)
Establish(Untimely-NOA)
Establish(NOA-Commencement)
Find(NOA-Commencement, Judgement-Date)
Find(Filed-by-Mail, Mailing-Date)
Establish(NOA-Filing-Date)
Find(Appeal-Filed, Filing-Date)
Establish(NOA-Due-Date)
Find(Appeal-Due, Due-Date = Mailing-Date + 45 Days)
Rule(Untimely-NOA)
Rule(Untimely-Appeal)
Rule(Authority)
Order(Show-Cause)
Order(Response-Due-Date = Order-Date + 14 Days)
Order(Sanction)
Order(Untimely-NOA-Rationale)
Organize(Show-Cause-Order)
Frame (Header)
Frame(Body)
Link(Preamble)
Link (Furthermore-Phrase)
Link (Show-Cause)
Frame(Footer)

Colorado Court of Appeals Order

No. 95CA0437 Tr. Ct. No. 91CV051

In re the Marriage of

Herbert W. Smythe

Appellant

and

Catherine Smythe

Appellee

To: Herbert W. Smythe, pro se appellant

From the notice of appeal filed by appellant and the register of actions submitted by the clerk of the district court, it appears that judgement was entered September 20, 1995 and mailed to counsel of record on September 22, 1995 and the notice of appeal was filed on November 7, 1995. Furthermore, it appears that the notice of appeal was due November 5, 1995. Thus, it appears that the notice of appeal was not timely filed.

See C.A.R. 4(a).

IT IS THEREFORE ORDERED that the appellant shall show cause, if any there is, in writing on or before December 18, 1995 why this appeal should not be dismissed with prejudice for failure to file a timely notice of appeal.

BY THE COURT

Date: December 4, 1995

Copies to: Counsel of Record

Figure 4. The illocutionary and rhetorical structure of *Smythe*

This example illustrates informally how a document grammar representing the illocutionary and rhetorical operators underlying a set of representative documents can be used to represent the illocutionary and rhetorical structures of new documents, which can in turn be used to generate the text of the document itself. A formal model of a document grammar for *Kirkpatrick* and *Canada* and a unification mechanism by which the text is realized from the resulting discourse structures is described in the next section.

4. An Implemented Document Planner

To investigate the computational mechanisms required of automated document planning for drafting judicial documents, we have designed and implemented a prototype document planning system. Given the facts of a case, the DOCU-PLANNER automatically creates show-cause orders that identify apparent defects and issue orders to appellants. The computational model on which the DOCU-PLANNER is based builds on a large body of work in computational linguistics on discourse generation. Just as discourse generators produce multi-sentential texts by creating hierarchical discourse structures, document planners can create documents in a similar fashion. By planning a discourse structure and then drafting a document that satisfies the illocutionary and rhetorical requirements dictated by this structure, the DOCU-PLANNER achieves the goals of accuracy, efficiency, and stylistic consistency.

The DOCU-PLANNER is implemented in a *unification-based* formalism (Elhadad, 1991). Its *document grammar* encodes the operators that are used to plan documents. In contrast to syntactic grammars which specify the structure of well-formed sentences, document grammars specify the structure of entire documents. In particular, they specify how the facts of a case are used to create illocutionary and rhetorical structures for a document to be generated for that case. Given a case, the system creates a document in two phases:

- *Document Planning:* The system unifies the facts of the case with the document grammar. This produces a discourse structure which is instantiated with the specifics of the given case.
- *Document Drafting:* The system interprets the resulting illocutionary and rhetorical specifications to create the final document in which the content, rhetorical organization, and formatting are completely specified.

The DOCU-PLANNER can operate in one of two modes. In *text* mode, it creates documents which are suitable for printing. In *web* mode, it produces documents that have been formatted in hyper-text markup language and can be displayed online with World Wide Web browsers.

This section is structured as follows. We first overview the theoretical foundations of document planning by discussing computational models of discourse planning. Next, we describe the DOCU-PLANNER's implementation of document grammars in the unification formalism. We then describe how the DOCU-PLANNER carries out document planning and document drafting. These are illustrated with a sample document creation session in which the system creates a show-cause order for a particular case.

4.1. FOUNDATIONS OF DISCOURSE GENERATION

The discourse-based approach to document planning builds on a strong foundation laid by computational linguists in discourse generation. Although for many years the primary focus of research in discourse structure was on accounting for the coherence of expository or other communicative text for purposes of understanding, e.g., (Grosz and Sidner, 1986b; Hobbs, 1979), discourse generation began to receive considerable attention beginning in the mid-1980s. Because document construction is inherently a discourse generation task, the representations that have been developed in the discourse generation community offer significant insights for designers of document planning systems. Computational models of discourse generation reason about the content and organization of knowledge to be communicated in order to automatically construct multi-sentential text.

To produce discourse automatically, well-represented discourse knowledge is crucial to the performance of discourse generators. *Discourse knowledge* is knowledge about how to perform content determination (determining the content of discourse that is being constructed) and organization (determining the structure of the discourse). The organizational aspect of discourse knowledge plays a particularly important role in the construction of extended discourse. *Discourse generation* is the process of applying discourse knowledge to produce multi-sentential or multi-paragraph texts. We can distinguish three approaches to discourse generation: schema-based approaches, plan-based approaches, and hybrid approaches. Beginning with work on schemata (McKeown, 1982; Paris, 1988), the field has matured over the past decade and a half to produce top-down discourse planners (Moore and Swartout, 1991; Suthers, 1991; Cawsey, 1992; Maybury, 1992; Hovy, 1993; Moore and Paris, 1993) and hybrid models (Suthers, 1991; Lester and Porter, 1997). We discuss each of these in turn.

The *schema-based* approach to discourse generation began with the pioneering dissertation of McKeown (McKeown, 1982), in which she analyzed naturally occurring texts to develop a set of schemata for describing concepts. *Schemata*, in this context, are ATN-like structures that represent patterns of discourse. For example, a schema for defining a concept includes instructions to identify its superclass, to name its parts, and to list its attributes. Schemata contain *rhetorical predicates*, e.g., “constituency,” which names the parts of an object. Each rhetorical predicate has an associated technique for extracting relevant propositions from a knowledge base. Schemata order the rhetorical predicates, some of which are optional, some of which can be repeated, and some of which can recursively invoke other schemata. The schema-based approach has been very influential in discourse generation. Although alternatives to schemata have since emerged, the schema-based approach dominated the field for several years because schemata successfully capture many aspects of discourse structure.

Schemata have been criticized because they lack flexibility. A top-down *planning* approach to discourse generation has been proposed to cope with this limited flexibility. This approach, which has dominated the field for the past few years (Moore and Swartout, 1991; Suthers, 1991; Cawsey, 1992; Maybury, 1992; Hovy, 1993; Moore and Paris, 1993), can be traced to Appelt’s work on planning referring expressions (Appelt, 1985), which in turn builds on earlier research on reasoning about speech acts in a planning paradigm (Cohen and Perrault, 1979). Planners offer a significant advantage over schema-based generators in that they can reason about the structure, content, and goals of explanations, as opposed to merely instantiating pre-existing plans embodied by schemata.

The operators of two seminal discourse planning systems are based on a theory of discourse known as Rhetorical Structure Theory (RST) (Mann and Thompson, 1987). At the heart of RST is the assumption that multi-sentential texts have an implicit structure that assists readers in assimilating the text's information. This assumption suggests that writers – and discourse generators – should provide this structure as they construct texts. RST was developed by analyzing a very large corpus of texts. It consists of a small number of rhetorical relations, e.g., Background, Motivation, and Sequence. By recursively applying the relations, one can parse a multi-sentential text into a tree where the leaves are clauses. Alternatively, a discourse generator can employ a top-down planner with RST-like operators to produce a multi-sentential text.

Perhaps the greatest problem faced by top-down planners is that their flexibility comes at the cost of an enormous search space arising from the combinatorics of multiple planning operators. The relative merits of schemata and top-down planners have been heavily debated. One conclusion is that schemata are best viewed as "fossilized discourse structures" that represented previously compiled plans (Hovy, 1993). To respond to a number of problems in discourse generation, Suthers developed a sophisticated *hybrid* approach that includes planning techniques as well as plan critics, simulation models, reorganization methods, and graph traversal (Suthers, 1991). By assembling these diverse mechanisms into a single architecture, he demonstrates how the complexities of discourse planning can be dealt with in a coherent framework. Lester and Porter developed the hybrid approach of *explanation design packages* (EDPs) for KNIGHT, a robust discourse generator for large-scale knowledge bases (Lester and Porter, 1996; Lester and Porter, 1997). KNIGHT's EDPs, which constitute a schema-like programming language for "discourse knowledge engineers," combine a hierarchical frame-based representation with embedded procedural constructs for knowledge-base access.

4.2. REPRESENTING DOCUMENT PLANNING KNOWLEDGE

Document planners can build on the large body of work in discourse generation. Because content determination problems and organization problems are analogous, many of the solutions that have been developed in discourse generation can be adapted for document planning. Discourse generators have been studied in genres that include expository texts, editorial texts, and advisory texts, inappropriate for document planners. Unlike expository texts, few documents have an exclusively communicative purpose. Rather, like editorial and advisory texts, documents are frequently concerned with *performative* utterances, a designation proposed by J.L. Austin (Austin, 1962) and later elaborated by others in speech act theory (Grice, 1975; Searle, 1969). Because individuals and institutions frequently draft documents to accomplish performative goals, such as creating or revoking legal, social, or institutional relationships, or eliciting information, representing the illocutionary structure of documents is critical.

Schema-based approaches and their derivatives, such as explanation design packages, have been most successful in producing expository texts. However, because schemata do not record intentional information and because document planners must be able to reason about the content of documents, a plan-based approach to representing the illocutionary structure underlying documents is more promising. Moore has observed that, "Any approach to discourse structure that relies solely on rhetorical relations or predicates and does not explicitly encode information about intentions is inadequate for handling dialogues" (Moore, 1995). For precisely the same reason, a pure schema-based approach that omits intentional knowledge would fare poorly in a document planning system.

Despite the great importance of illocutionary knowledge, the rhetorical structure of documents is equally critical. For example, conforming to the precise structure and formatting conventions of court documents is indispensable to the success of document planners for judicial applications. Hence, we must encode rhetorical knowledge as well, perhaps in a schema-based formalism. However, adopting a hybrid model that employs two formalisms – a plan-based approach for illocutionary knowledge and a schema-based approach for rhetorical knowledge – is cumbersome. We therefore opt for a uniform approach and are presented with two alternatives: we can either embed illocutionary and rhetorical knowledge in the same operators – to some extent, this is the RST approach – or we can somehow decouple them. Decoupling offers two important advantages:

- *Increased fidelity of representation*: Decoupling enables the illocutionary and rhetorical theories to reflect the reality that, in many document planning tasks, the illocutionary structure and rhetorical structure do not stand in a subsumptive relationship. This phenomenon seems to have arisen because the rhetorical structure of documents has taken on a life of its own, spawning discourse organization conventions that are sometimes orthogonal to the illocutionary structure.
- *Increased ease of inspection:* Decoupling the structures permits users to view either the full illocutionary structure or the rhetorical structure in isolation from the other. If the two sets of operators co-exist separately, users can request an illocutionary view of the document separate from the rhetorical structure, and *vice versa*.

The discourse-based approach encodes knowledge about illocutionary and rhetorical structures in a uniform, plan-based representation. A *document grammar*, which encodes illocutionary and rhetorical operators, can be represented in a functional unification grammar formalism (Kay, 1979). Given the facts of a specific case, the document planner sub-goals on the illocutionary operators to create an illocutionary structure for the document while it sub-goals on the rhetorical operators to create a rhetorical structure for the document.

Illocutionary operators represent the top-down decomposition of intentional goals into sub-goals. For example, the goal of the operator

```
Establish(Jurisdictional-Defect) ⟹
        Establish(Appeals)
        At-Least-One [ Establish(Untimely-Notice-Of-Appeal),
                       Establish(Subject-Defect),
                       Establish(Finality-Defect) ]
```

is to establish a jurisdictional defect. It accomplishes this by attempting to establish what orders are being appealed from (*i.e.*, Establish(Appeals)) and then, for each such order, attempting to establish as many of the following as possible: untimely notice of appeal, a subject defect, and/or a finality defect. If it is able to achieve the goal of establishing appeals as well as at least one of the other three sub-goals, it succeeds; otherwise it fails.

To further illustrate, consider the operator that establishes an untimely notice of appeal:

```
Establish(Untimely-Notice-Of-Appeal) ⟹
        IF <judgment-date>
          { Establish(NOA-Commencement)
            Establish(<filing-date>, Notice-Of-Appeal)
            Establish(<due-date>, Notice-Of-Appeal)
            Rule(Untimely-Notice-Of-Appeal) }
```

Five aspects of this operator are noteworthy. First, the goal of the operator appears as a sub-goal in the preceding operator. As a result, when the system attempts to satisfy the first operator which establishes a jurisdictional defect, the second operator is invoked to establish an untimely notice of appeal. Second, the system inspects its knowledge of the case to determine if there is a judgment date. If there is one, it proceeds to the sub-goals; if there is not, it fails. Third, the sub-goals are conjunctive: all of them must be achieved for the operator to succeed. The sub-goal Rule(Untimely-Notice-Of-Appeal) succeeds only if the <filing-date> is later than the <due-date> Fourth, just as in logic programming, variable bindings must be made consistently for the goal to be achieved. Finally, text-emission goals are present in the Rule goal. If the illocutionary plan containing an instantiated version of this operator ultimately succeeds, then text emission will occur.

Rhetorical operators represent the top-down decomposition of textual organization goals into sub-goals. For example, the operator

```
Organize(Show-Cause-Order) ⟹ Frame(Header)
                              Frame(Body)
                              Frame(Footer)
```

organizes the global structure of the major sections of documents. Some rhetorical operators impose an organization on a particular section, e.g.,

```
Frame(Body) ⟹ Link(Preamble)
              Link(However-Phrase)
              Link(Furthermore-Phrase)
              Link(Thus-Phrase)
              Link(Show-Cause)
```

In addition to the illocutionary and rhetorical operators, the document grammar also includes organizational constraints that specify the interleaving of operators that is required to produce the final instantiated discourse structure. For example, the constraint

```
Sequence(Preqs-For-Dismissal) = Left-To-Right [ Frame(Header),
                                                Link(Preamble),
                                                Establish(Jurisdictional-Defects),
                                                Order(Show-Cause),
                                                Frame(Footer) ]
```

specifies how achieving the illocutionary goals of establishing jurisdictional defects and ordering the appellant to show cause should be interleaved with the rhetorical subgoals of constructing a a header, preamble, and footer.

Our document grammar is organized into *operator modules*, each of which contains a collection of operators that perform a similar function. The DOCU-PLANNER currently contains 7 modules with 85 operators (Figure 5). The *Establishment*, *Ordering*, *Findings*, and *Rulings* modules contain the illocutionary operators. The *Organizational and Constraint* and *Framing and Linking* modules contain the rhetorical operators. The *Informing* module contains operators that emit text; these are invoked as sub-goals of both illocutionary and rhetorical operators. Modularizing the document grammar in this fashion facilitates construction of new operators and maintenance of existing operators.

Our operators are represented in a functional unification formalism. For example, the four operators discussed above are represented as shown in Figure 6. The syntax of the formalism is that of functional descriptions (Elhadad, 1991), which is defined recursively: a functional description consists of a list of pairs, where the second item in each pair is either atomic or is itself a functional description. Collectively, the functional descriptions for the illocutionary and rhetorical operators define the document grammar. Employing the unification formalism significantly increases the planner's flexibility by enabling it to create documents for an enormous variety of case facts.

4.3. DOCUMENT CREATION

The DOCU-PLANNER (Figure 7) is a unification-based implementation of a planning approach to document construction. Given the facts of an appellate case, the DOCU-PLANNER constructs show-cause orders with the appropriate content, organization, and stylization. It consists of three components:

- **Document Grammar**: Encodes the illocutionary and rhetorical operators.
- **Document Planner**: Constructs discourse structures (linked and instantiated illocutionary and rhetorical operators).
- **Document Drafter**: Creates completed documents by traversing discourse structures and emitting formatted text.

(1) **The Establishment Module**
Function: Creates backbone of illocutionary structure
Type: Illocutionary
Example: `Establish-Untimely-Notice-Of-Appeal`
Number of operators: 12

(2) **The Ordering Module**
Function: Enunciates performative text
Type: Illocutionary
Example: `Sanction`
Number of operators: 10

(3) **The Findings Module**
Function: Makes factual findings
Type: Illocutionary
Example: `Judgment`
Number of operators: 12

(4) **The Rulings Module**
Function: Makes legal ruling
Type: Illocutionary
Example: `Non-Final-Order`
Number of operators: 7

(5) **The Organizational and Constraint Module**
Function: Imposes rhetorical organization
Type: Rhetorical
Example: `Sequence(Show-Cause)`
Number of operators (and constraints): 8

(6) **The Framing and Linking Module**
Functions: Creates boiler plate text including header and footer
Creates linking phrases
Creates specialized formatting directives
Type: Rhetorical
Example: `Frame(Header)`
Number of operators: 19

(7) **The Informing Module**
Functions: Emits and conjoins pre-computed text segments
Creates inter-paragraph formatting directives
Type: Text Emission
Example: `Inform-Appeal-Filed`
Number of operators: 17

Figure 5. Operator modules of the DOCU-PLANNER's document grammar

```
((type jurisdictional-defect)
 (untimely-defect ((alt untimely-defect
                        (((cat establish)
                          (type untimely-noa))
                         ((cat stop)
                          (untimely-noa? no))))))
 (subject-defect ((alt subject-defect
                       (((cat establish)
                         (type subject-matter-defect))
                        ((cat stop)
                         (subject-matter-defect? no))))))
 (finality-defect ((alt finality-defect
                        (((cat establish)
                          (type finality-defect))
                         ((cat stop)
                          (finality-defect? no)))))))

((type untimely-noa)
 (date-later-than #(external due-date))
 (date-later-than yes)
 (e-due-date ((cat establish)
              (type noa-due-date)))
 (e-filing-date ((cat establish)
                 (type noa-filing-date)))
 (and ((cat text) (lex "and")))
 (f-filing-date ((cat find)
                 (type noa-filing-date)))
 (furthermore ((cat text) (lex "Furthermore, it appears that")))
 (f-due-date ((cat find)
              (type noa-due-date)))
 (thus ((cat text) (lex "Thus, it appears that")))
 (rule-untimely ((cat rule)
                 (type untimely-noa)))
 (authority {authority}))

 (pattern (untimely-defect subject-defect finality-defect))

 (pattern (appeal-filed filing-date end-sent)
```

Figure 6. Unification formalism of illocutionary and rhetorical operators

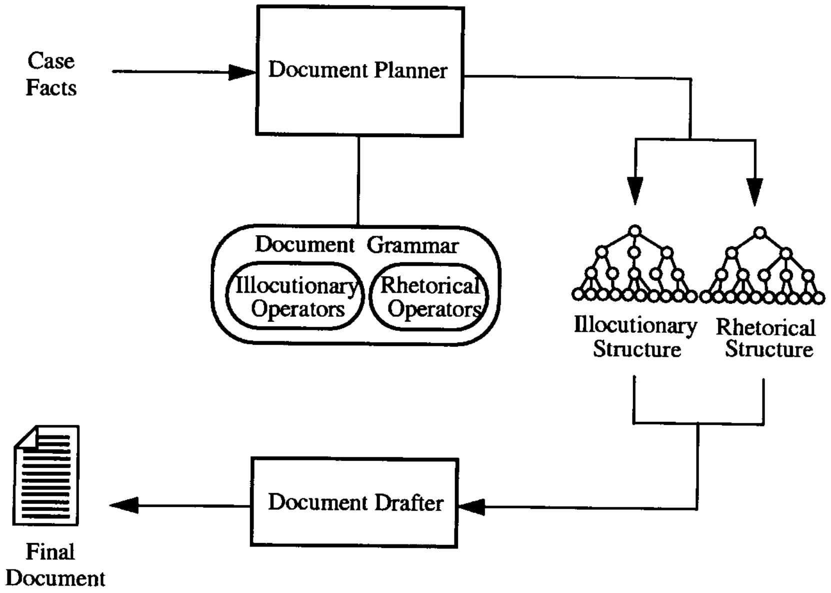


Figure 7. The DOCU-PLANNER's architecture

Given the specifics of a particular case, the document planner backchains on the illocutionary operators in a problem-decomposition fashion to construct the evolving document's illocutionary structure. Similarly, it backchains on the rhetorical operators to construct the document's rhetorical structure. Both of these tasks are accomplished simultaneously through unification of the representation of the case facts with the document grammar. The net result of this computation is a discourse structure in which the operators defining the illocutionary and rhetorical structures are fully instantiated and linked together through variable bindings. Many nodes in these structures specify the production of text segments and formatting directives. Next, the document drafter conducts a pre-order traversal of the discourse structure produced by the document planner. It then concatenates the text obtained from this traversal and embeds formatting directives (which were also specified in the discourse structure) in the concatenated text. Finally, it interprets the resulting linear structure, thereby creating the completed document.

The DOCU-PLANNER is implemented with FUF (Functional Unification Formalism), a robust unification environment developed at Columbia University (Elhadad, 1991; Elhadad, 1992). FUF is itself implemented in Lisp, as is the DOCU-PLANNER's drafting system. The entire system runs in Harlequin Lisp on a DEC Alpha. Document creation is accomplished very quickly. Given the case facts, the typical amount of time to create an appellate jurisdictional show cause order is less than one second.

To illustrate the DOCU-PLANNER's behavior, consider *Smythe vs. Smythe*. Recall that in *Smythe* Herbert Smythe filed a notice of appeal on November 7, 1995 in

```
Case: Smythe
Notification-Method: Mailing
Appellant: Herbert W. Smythe
Appellee: Catherine Smythe
Substantive-Legal-Area: Domestic
Judgment-Date:
               Month: 9
               Day: 20
               Year 1995
Filing-Date:
               Month: 11
               Day: 7
               Year 1995
Mailing-Date:
               Month: 9
               Day: 22
               Year 1995
Order-Date:
               Month: 12
               Day: 4
               Year 1995
Case-Number: No. 95CA0437
Order-Number: Tr. Ct. No. 91CV051
Authority: See C.A.R. 4(a).
```

Figure 8. Representation of case facts of *Smythe vs. Smythe*

response to a judgment that was entered on September 20, 1995. The summary judgment for this domestic case was mailed on September 22. The DocuPlanner's representation of the facts of *Smythe*, including the cited authority, the appellant, and the appellee, are represented in Figure 8.

Given these facts, the document planner first creates a discourse structure for a show-cause order by unifying the input representation with the document grammar. By unifying the top-level illocutionary operators with the input data, then backchaining on these instantiated operators and repeating this process recursively, the document planner constructs a fully instantiated illocutionary structure. Beginning with the goal of establishing the prerequisites for dismissal, which is included in the findings of the case, the document planner posts two sub-goals, establishing jurisdictional defects and ordering the appellant to show cause. It then attempts to achieve each of these in turn. To establish jurisdictional defects, the planner first attempts to establish an untimely notice of appeal. To accomplish this sub-goal, it attempts to establish that the due date for the notice of appeal preceded the filing

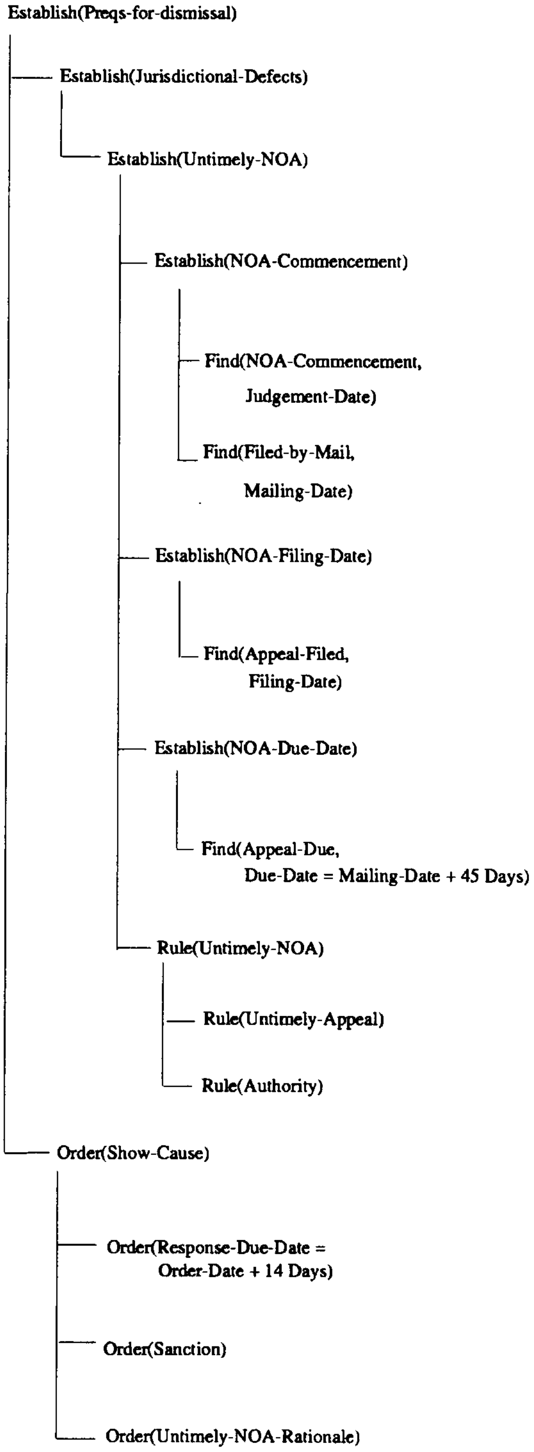

Establish(Preqs-for-dismissal)
Establish(Jurisdictional-Defects)
Establish(Untimely-NOA)
Establish(NOA-Commencement)
Find(NOA-Commencement, Judgement-Date)
Find(Filed-by-Mail, Mailing-Date)
Establish(NOA-Filing-Date)
Find(Appeal-Filed, Filing-Date)
Establish(NOA-Due-Date)
Find(Appeal-Due, Due-Date = Mailing-Date + 45 Days)
Rule(Untimely-NOA)
Rule(Untimely-Appeal)
Rule(Authority)
Order(Show-Cause)
Order(Response-Due-Date = Order-Date + 14 Days)
Order(Sanction)
Order(Untimely-NOA-Rationale)

Figure 9. The illocutionary structure constructed for *Smythe*

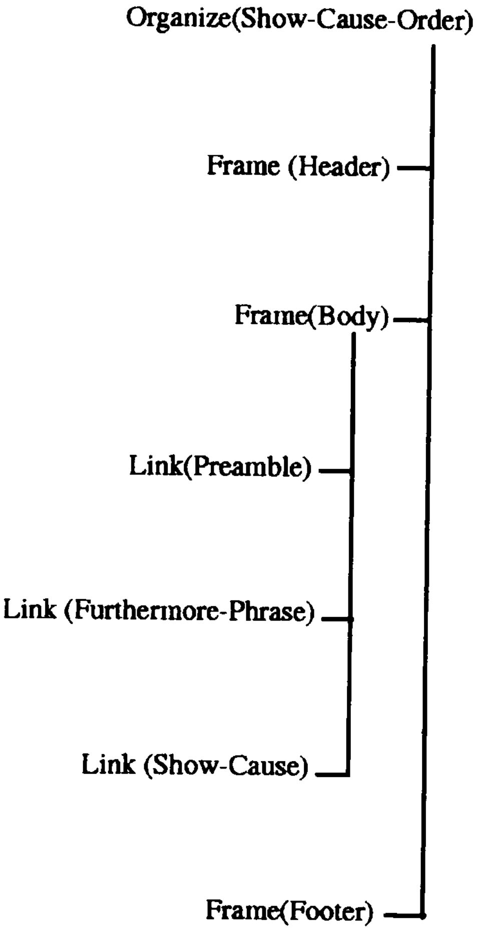


Figure 10. The rhetorical structure of *Smythe*.

date of the appeal. By analyzing the date information contained in the facts of the case, it determines that the goal is satisfied.

Next, it posts four sub-goals which, if conjunctively satisfied, will achieve the goal of demonstrating that the appeal was untimely: (1) establish the date of commencement for the notice of appeal (NOA); (2) establish the filing date of the NOA; (3) establish the due date for the NOA; and (4) make a ruling that the NOA was untimely. To achieve subgoal (1), it first makes a finding of the date of commencement based on the information in the case data. It then attempts to make a finding that the commencement date is the same as the judgment date, which is stated in the case data. However, in order for the judgment date to be the same as the commencement date, the appellant must have been present at judgment. The case data contradicts this proposition, thereby causing this sub-goal to fail. It then attempts to find another means for determining the commencement date. This is accomplished by attempting to apply an alternative illocutionary operator with the same goal. An alternative operator it considers is the findings operator whose argument is `notification-by-mail`. This unifies successfully with the case data. As a result, sub-goal (1) noted above is achieved. The sibling sub-goals (2), (3), and (4) are then achieved in a similar manner. The net result of this process is the illocutionary structure shown in Figure 9.

The document planner creates the rhetorical structure in a similar way. By unifying rhetorical operators with the input data, backchaining on these instantiated operators, and recursing, the document planner constructs a fully instantiated rhetorical structure, which also includes formatting directives. First, the top-most rhetorical operator posts three sub-goals: (1) construct a "header" which includes case specific details such as the names of the appellant and appellee and court number; (2) construct the body of the document; and (3) construct the "footer" which includes additional case specifics such as the order date. Planning the rhetorical structure for the body involves emitting the preamble, connective phrases such as 'furthermore it appears that', and other boilerplate phrases. The net result of backchaining on rhetorical operators is the rhetorical structure shown in Figure 10.

To enforce consistency of constraints across instantiations of illocutionary and rhetorical operators, the planner employs a global approach to unification in which variable bindings in the rhetorical structure are made consistent with bindings in the illocutionary structure. Illocutionary planning and rhetorical planning are interleaved at runtime. As variables in the operators are bound to the specifics of the case, the constraints they impose on the instantiation of other operators are propagated throughout the grammar to create the discourse structure. When document planning is complete, the discourse structure contains both the intentional inferences supporting the findings and also the discourse inferences supporting the organizational and formatting decisions. The details of the discourse structure produced by the document planner for *Smythe* are shown in Appendix B.

The document drafter performs its work in two distinct phases: traversal and interpretive concatenation. First, it conducts a pre-order traversal of the discourse structure. For each leaf of the discourse structure, it determines if a text segment has already been constructed for the node by Inform operators. If none is found, it must create a text segment that expresses the content in the node. Inform operators can specify the inclusion of a noun phrase (e.g., 'the trial court's order granting summary judgment in favor of the plaintiff'), a verb phrase (e.g., 'The notice of appeal was filed on'), a connective phrase (e.g., 'as to'), a sentence (e.g., 'See C.A.R. 4(a).'), a formatting directive (e.g., a paragraph break), or a reference to the facts of a case (e.g., <appellant>). To create a text segment, the drafter examines the node content and constructs a phrase that expresses it. For example, to realize a date, it extracts the month, day, and year features and transforms them into a date phrase, (*e.g.*, 'March 27, 1987').

During the interpretive concatenation phase, the document drafter examines each object produced during the traversal, which includes both text segments and formatting directives. By concatenating the text segments with the interpreted formatting directives in the order specified by the pre-order traversal of the discourse structure, the drafter produces the final document.[6] For example, the rhetorical structure created for *Smythe* includes paragraph breaks and right justifications that conform to the conventions of show-cause orders issued by the Colorado Court of Appeals. Figure 4 displays the resulting show-cause order flanked by the illocutionary and rhetorical structures that produced it.

The DOCU-PLANNER creates documents efficiently. For example, creating the *Smythe* show-cause order required 775 milliseconds on a DEC Alpha and *Canada* required 937 milliseconds. Typically, document planning *per se* consumes approximately three quarters of the total execution time while document drafting takes approximately one quarter of the time.

5. Related Work

Our approach to automated document generation draws on four different lines of research: discourse structure analysis, the theory of argumentation, explanation generation, and automated document drafting. The primary focus of research in discourse structure has been accounting for the coherence of expository or other communicative text through hierarchical structures of rhetorical and other discourse relations, e.g., (Grosz and Sidner, 1986a; Hobbs, 1979). The formalization of inter-sentential discourse relations is a key requirement for the development of automated document generation systems.

The most directly relevant portion of research in discourse structure is speech act theory. Initiated by J.L. Austin, who was primarily concerned with explicit performatives (Austin, 1962), speech act theory addresses the illocutionary content of discourse, i.e., the goals that a speaker intends to accomplish through that discourse (Grice, 1975; Searle, 1969).

The theory of argumentation addresses texts intended to persuade, establish, or prove. For example, Toulmin (Toulmin, 1958) analyzed argumentative texts in terms of the concepts of warrant, ground, conclusion, backing, and qualification. This model has been widely applied to the analysis (Marshall, 1989; Zeleznikow and Stranieri, 1995) and creation (Bench-Capon and Staniford, 1995) of legal documents. Argument structure, like other forms of illocutionary goal structure but unlike rhetorical structure, does not directly address the "surface" form of texts. This line of research is particularly relevant to the analysis of the illocutionary structure of persuasive or dispositive documents, such as legal briefs and judicial decisions (Branting, 1993a).

The explanation community has extensively studied the process of planning and realizing text given a set of discourse specifications. Over the past decade, their

[6] If the user has requested *web* mode, the formatting directives are first translated to hyper-text markup language (HTML) formatting commands, which collectively produce a properly formatted online document.

research on discourse planning (McKeown, 1985; Paris, 1988; Hovy, 1990; Hovy, 1993; Cawsey, 1992; Suthers, 1993; Moore, 1995; Mittal, 1993; Lester and Porter, 1996) has produced a variety of techniques for determining the content and organization of many genres of text. Perhaps because of the necessity of coping with the myriad underlying rhetorical, illocutionary, and argument structures in discourse generation, this work has yielded a variety of mechanisms for determining the content and organization of multi-sentential text, a key capability of self-explaining documents.

Automated document drafting research is the fourth relevant research area. Two important areas of automated document drafting research are automated legal drafting and automated report generation. A large number of automated legal drafting systems have been developed in recent years, but most involve creation of text templates that are then instantiated to create particular documents (Lauritsen, 1992). This approach has been successfully applied to automated drafting of highly predictable, regular documents (Spirgel-Sinclair, 1988).

Some progress has been made in exploiting explicit representations of the relationship between generic documents and document instances and of constraints among document components (Daskalopulu and Sergot, 1995). However, there is a growing recognition in the Law and AI community that a declarative representation of the knowledge underlying the selection and configuration of textual elements is essential for the development of tools that embody the expertise of legal drafting experts (Gordon, 1989; Lauritsen, 1993).

Several recent systems have used declarative representations of legal rules but only partially declarative representations of rhetorical structure. For example, JEDA (Pethe et al., 1989) used a declarative representation of legal rules, but mediated document construction entirely through procedural rules. Similarly, LAW CLERK (Branting, 1993a) used an explicit representation of legal rules and a simple record structure for administrative law decisions. LAW CLERK instantiated and wrote text templates associated with predicate/truth-value pairs to the fields of decision record during back-chaining.

A more detailed declarative rhetorical model was used in PLAID (Bench-Capon and Staniford, 1995), which produced a document by generating an illocutionary structure whose nodes were tagged according to their role in the argument (claim, rebuttal, support, qualification, *etc.*). This structure was then pruned to exclude premises which should be implicit in the final presentation and organized into a structure, including linking text, based on a high-level rhetorical template.

The applied computational linguistics community has addressed the task of automated report generation from an underlying domain structure. Kittredge *et al.* have observed that representing new domain-dependent discourse knowledge – they term it “domain communication knowledge” – is required to create advanced report generators, e.g., for special purpose report planning (Kittredge et al., 1991). Given a representation of a particular domain for a particular application, knowledge-based report generation is the task of automatically producing clearly stated reports that are relevant to users of the application. This community has focused its efforts on deriving technical documentation from program traces generated during software development or use (Korelsky et al., 1993; Johnson, 1994; McKeown et al., 1995) and on producing customized patient information reports for medical applications (DiMarco et al., 1995).

6. Discussion and Future Work

In this paper we have presented a model of the illocutionary and rhetorical structures underlying a representative type of judicial documents – jurisdictional show-cause orders – and have shown how these structures can be used to form a document grammar that can generate new documents using a unification-based procedure. In Section 1 we argued that high standards of correctness and consistency are essential in judicial document drafting, and in Section 2 we argued for the importance of increased efficiency in drafting routine judicial documents. We believe that document grammars provide an extremely powerful method for simultaneously achieving correctness, consistency, and efficiency.

The strength of the discourse structure representation of judicial documents is that it explicitly represents *both* an illocutionary structure, which arises from the applicable domain legal rules, and a rhetorical structure, which expresses the stylistic and discourse conventions of the genre. Earlier approaches to routine judicial document drafting generally omitted one aspect of the discourse structure or conflated the two.

We advocate an approach that uses explicit representations of illocutionary and rhetorical structures in a document grammar that applies a unification-based formalism to this grammar to create documents. There are two key advantages to this approach. First it aids knowledge acquisition and system building because it facilitates a conceptual model that cleanly separates (1) substantive domain rules, e.g., the requirements for jurisdiction, (2) genre-specific stylistic and discourse conventions, and (3) the procedure for creating documents that conform to the domain rules and genre conventions. Illocutionary operators can be conceptualized in terms of the underlying legal rules, without consideration of the way in which the rules will be embedded in a document. Similarly, rhetorical operators can be formalized independently of the legal rules that the document will ultimately express. Both forms of operators can be developed without requiring consideration of the procedures governing their use in document drafting.

A second, related, advantage is that this approach assists validation (because each component can be tested separately) and maintenance (because either the illocutionary or the rhetorical operator set can be modified without affecting the other). For example, a change in jurisdictional rules can, in general, be accommodated by modifying the illocutionary operators without requiring any changes in the rhetorical operators.

This paper has focused on document drafting *ab initio* given a relatively complete document grammar. We believe that the dual-justification structure also has important potential benefits for retrieval, explanation, and adaptation of existing documents, comparison of alternative drafts of documents at a “deep” illocutionary level, and maintenance of multi-generation documents. Document drafting through reuse is a promising method of extending the coverage of a document grammar, since retrieval and adaptation can be performed with even an incomplete set of illocutionary operators. Accordingly, the first step in our research agenda is developing mechanisms that use discourse structure for retrieval, explanation, comparison, and interactive adaptation of existing documents.

This paper has illustrated the creation of new documents from a document grammar that was manually constructed. However, constructing a document grammar is a challenging activity that currently requires considerable knowledge of the underlying unification formalism. Therefore, the second step in our current research agenda is to develop a semi-automated knowledge acquisition tool for interactively acquiring document grammars. We envision that domain experts, beginning with a core document grammar, will use this tool to construct document grammars that exhibit increasingly greater coverage. However, rather than expressing the document grammar directly in the syntactically complex unification formalism, the grammar acquisition tool will enable them to state illocutionary and rhetorical operators in a high-level language and help them rapidly evaluate the coverage of the grammar until it generates precisely the set of documents they intend.

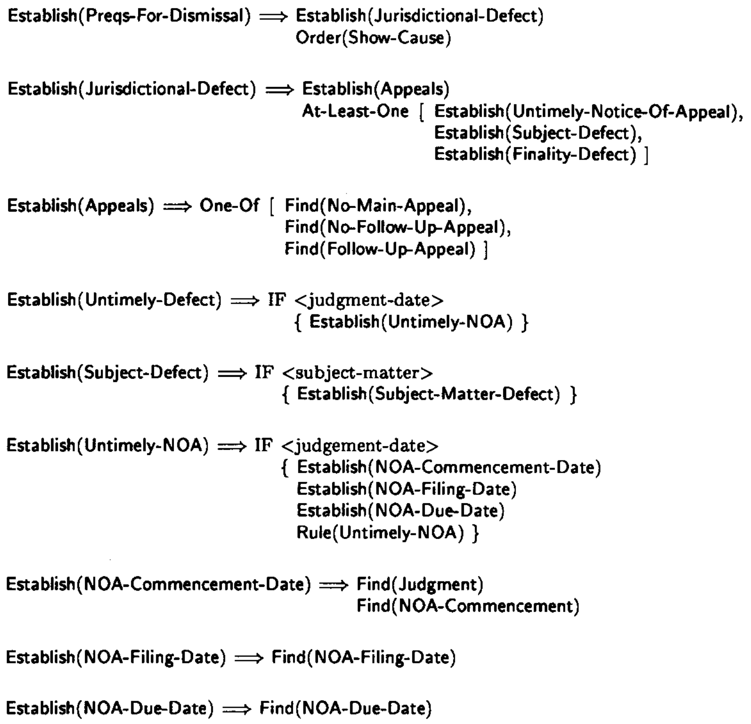
7. Summary

This paper has presented a model of document structure that makes explicit (1) the document’s illocutionary structure, i.e., the connection between the document drafter’s goals and the text intended to achieve those goals, and (2) the document’s rhetorical structure i.e., the stylistic and discourse conventions of the document’s genre. This model was applied to a representative class of judicial orders, jurisdictional show-cause orders. The ability of a document grammar based on the illocutionary and rhetorical structure of representative documents to synthesize additional documents was then illustrated with a simple example. Practical knowledge-based judicial document drafting systems for routine judicial documents would make a significant contribution to judicial efficiency. The approach described in this paper is a first step towards the development of such practical systems.

Appendix A

THE COMPLETE DOCUMENT GRAMMAR

This appendix presents the document grammar used to synthesize the show cause orders for *Smythe*, *Kirkpatrick*, and *Canada*. For clarity, the grammar is expressed in a context-free-style syntax; it is in fact implemented in the functional unification language, FUF, whose syntax is significantly more complex. Expressions on the left-hand side of each production are *predicate* non-terminals, e.g., Rule(Untimely-Notice-Of-Appeal), while the right-hand side consists of one or more combinations of (1) a predicate non-terminal, (2) a *collective* non-terminal e.g., At-Least-One[X, Y, Z], (3) a *conditionalized* non-terminal, e.g., If{*condition*}, and (4) a variable, which is indicated with braces, e.g., <*var*>. The *null* expression is indicated with ϵ. Ordering constraints, which are represented with Sequence predicates, are after the productions.



```
Establish(Preqs-For-Dismissal) ⟹ Establish(Jurisdictional-Defect)
                                  Order(Show-Cause)

Establish(Jurisdictional-Defect) ⟹ Establish(Appeals)
                                    At-Least-One [ Establish(Untimely-Notice-Of-Appeal),
                                                   Establish(Subject-Defect),
                                                   Establish(Finality-Defect) ]

Establish(Appeals) ⟹ One-Of [ Find(No-Main-Appeal),
                               Find(No-Follow-Up-Appeal),
                               Find(Follow-Up-Appeal) ]

Establish(Untimely-Defect) ⟹ IF <judgment-date>
                               { Establish(Untimely-NOA) }

Establish(Subject-Defect) ⟹ IF <subject-matter>
                              { Establish(Subject-Matter-Defect) }

Establish(Untimely-NOA) ⟹ IF <judgement-date>
                            { Establish(NOA-Commencement-Date)
                              Establish(NOA-Filing-Date)
                              Establish(NOA-Due-Date)
                              Rule(Untimely-NOA) }

Establish(NOA-Commencement-Date) ⟹ Find(Judgment)
                                    Find(NOA-Commencement)

Establish(NOA-Filing-Date) ⟹ Find(NOA-Filing-Date)

Establish(NOA-Due-Date) ⟹ Find(NOA-Due-Date)
Establish(Finality-Defect) ⟹ IF <order-appealed-from>
                              { Establish(Order-Appealed-From)
                                Establish(Non-Final-Order) }

Establish(Non-Final-Order) ⟹ Rule(Non-Final-Order)

Establish(Order-Appealed-From) ⟹ Find(Appeal-Source)
                                  <order-appealed-from>

Inform(Appeal-Due) ⟹ "the notice of appeal was due"

Inform(One-Appeal-Only) ⟹ "judgment was entered"

Inform(Main-Appeal-Entered) ⟹ "was entered"

Inform(Appeal-Source) ⟹ "this appeal is from"

Inform(Untimely-Appeal) ⟹ "the notice of appeal was not timely filed."

Inform(Untimely-Main-Appeal) ⟹ "the notice of appeal was not timely"

Inform(Untimely-NOA-Rationale) ⟹ "for failure to file a timely notice of appeal."

Inform(Appeal-Filed) ⟹ "the notice of appeal was filed on"

Inform(Final-Sanction) ⟹ "why this appeal should not be dismissed with prejudice"

Inform(Non-Final-Sanction) ⟹ "why this appeal should not be dismissed without prejudice"

Inform(Partial-Final-Sanction) ⟹ "why this appeal should not be partially dismissed with
                                   prejudice to the extent that defendant seeks review of"

Inform(Partial-Non-Final-Sanction) ⟹ "why this appeal should not be partially dismissed
                                       without prejudice to the extent that defendant seeks
                                       review of"

Inform(Non-Final-Judgment) ⟹ "this is not a final judgment because it does not end
                               'the particular action in which it is entered, leaving
                               nothing further for the court pronouncing it to do in
                               order to completely determine the rights of the parties
                               involved in the proceeding.' "

Inform(Appeal) ⟹ "defendant is appealing from"

Inform(Appeal-Both) ⟹ "defendant is appealing from both"

Inform(Non-Final-Order-Rationale) ⟹ "for failure to file a final appealable order."

Inform(Mailed) ⟹ "and mailed to counsel of record on"

Order(Show-Cause) ⟹ Order(Response-Due-Date)
                     One-Of [ Order(Sanction),
                              Order(Partial-Sanction) ]
                     One-Of [ Order(Non-Final-Order-Rationale),
                              Order(Untimely-NOA-Rationale) ]

Order(Sanction) ⟹ IF NOT <main-appeal>
                   { One-Of [ Order(Non-Final-Sanction),
                     Order(Final-Sanction) ] }
Order(Partial-Sanction) ⟹ IF <main-appeal>
                             { One-of [ Order(Partial-Non-Final-Sanction),
                                        Order(Partial-Final-Sanction) ]
                               <main-appeal> }

Order(Non-Final-Order-Rationale) ⟹ IF <order-appealed-from>
                                      { Inform(Non-Final-Order-Rationale) }

Order(Untimely-NOA-Rationale) ⟹ IF NOT <order-appealed-from>
                                   { Inform(Untimely-NOA-Rationale) }

Order(Response-Due-Date) ⟹ Compute-Response-Date(<judgment-date>, <mailing-date>)

Order(Non-Final-Sanction) ⟹ IF <order-appealed-from>
                               { Inform(Non-Final-Sanction) }

Order(Final-Sanction) ⟹ IF NOT <order-appealed-from>
                           { Inform(Final-Sanction) }

Order(Partial-Non-Final-Sanction) ⟹ IF <order-appealed-from>
                                       { Inform(Partial-Non-Final-Sanction) }

Order(Partial-Final-Sanction) ⟹ IF NOT <order-appealed-from>
                                   { Inform(Partial-Final-Sanction) }

Find(Judgment) ⟹ One-Of [ Find(One-Appeal-Only),
                           Find(Main-Appeal) ]

Find(One-Appeal-Only) ⟹ IF NOT <main-appeal>
                           { Inform(One-Appeal-Only) }

Find(Main-Appeal) ⟹ IF <main-appeal>
                       { Inform(Main-Appeal-Entered) }

Find(NOA-Commencement) ⟹ One-Of [ Find(Mailed),
                                   Find(Present) ]

Find(Mailed) ⟹ IF EQUAL(<notification-method>, mailing)
                  { <judgment-date>
                    Inform(Mailed)
                    <mailing-date> }

Find(Present) ⟹ IF EQUAL(<notification-method>, present-at-judgment)
                   { <judgment-date> }

Find(No-Main-Appeal) ⟹ IF NOT <main-appeal>
                          { Inform(Appeal-Source)}

Find(No-Follow-Up-Appeal) ⟹ IF NOT <follow-up-appeal>
                               { Inform(Appeal)
                                 <main-appeal> }

Find(Followup-Appeal) ⟹ IF <followup-appeal>
                           { Inform(Appeal-Both)
                             <main-appeal>
                             <follow-up-appeal> }

Find(NOA-Due-Date) ⟹ Inform(Appeal-Due)
                      Compute-Due-Date(<judgment-date>, <mailing-date>)

Find(Appeal-Source) ⟹ Inform(Appeal-Source)
Find(NOA-Filing-Date) ⟹ Link(And)
                        Inform(Appeal-Filed)
                        <filing-date>

Rule(Untimely-NOA) ⟹ One-Of [ Rule(No-Main-Appeal),
                              Rule(Main-Appeal) ]
                     Rule(Authority)

Rule(No-Main-Appeal) ⟹ IF NOT <main-appeal>
                          { Rule(Untimely-Appeal) }

Rule(Main-Appeal) ⟹ IF <main-appeal>
                       { Rule(Untimely-Main-Appeal) }

Rule(Untimely-Main-Appeal) ⟹ Inform(Untimely-Main-Appeal)
                             <main-appeal>

Rule(Untimely-Appeal) ⟹ Inform(Untimely-Appeal)

Rule(Non-Final-Order) ⟹ Inform(Non-Final-Judgment)

Rule(Authority) ⟹ <authority>

Frame(Header) ⟹ "Colorado Court of Appeals                  Order"
                <case-number>
                <order-number>
                Frame(Referent)
                <appellant>
                "                                   Appellant"
                "and"
                <appellee>
                "                                   Appellee"
                "To: "
                <appellant>
                Frame(Attorneys)

Frame(Referent) ⟹ One-Of [ Frame(Domestic),
                           Frame(Civil) ]

Frame(Domestic) ⟹ IF EQUAL(<substantive-legal-area>, domestic)
                     { "In re the Marriage of" }

Frame(Civil) ⟹ IF EQUAL(<substantive-legal-area>, civil)
                  { <attorneys> }

Frame(Attorneys) ⟹ One-Of [ Frame(Not-Pro-Se),
                            Frame(Pro-Se) ]

Frame(Not-Pro-Se) ⟹ IF <attorneys>
                       { <attorneys> }

Frame(Pro-Se) ⟹ "pro se appellant"

Frame(Body) ⟹ Link(Preamble)
              Link(However)
              Link(Furthermore)
              Link(Thus)
              Link(Show-Cause)

Frame(Body) ⟹ Link(Preamble)
              Link(Further)
              Link(Show-Cause)
```

```
Frame(Body) ⟹ Link(Preamble)
               Link(Furthermore)

Frame(Footer) ⟹ " BY THE COURT"
                 "Date: "
                 <order-date>
                 "Copies to: Counsel of Record"

Link(Preamble) ⟹ "From the notice of appeal filed by appellant and the
                  register of actions submitted by the clerk of the
                  district court, it appears that"

Link(Thus) ⟹ "Thus, it appears that"

Link(Further) ⟹ "It further appears that"

Link(Furthermore) ⟹ "Furthermore, it appears that"

Link(Show-Cause) ⟹ "IT IS THEREFORE ORDERED that the appellant shall
                     show cause, if any there is, in writing on or before"

Link(However) ⟹ "However, it appears that"

Link(As-To) ⟹ "as to"

Link(And) ⟹ " and"

Organize(Show-Cause-Order) ⟹ Frame(Header)
                              Frame(Body)
                              Frame(Footer)

Sequence(Preqs-For-Dismissal) = Left-To-Right [ Frame(Header),
                                                Link(Preamble),
                                                Establish(Jurisdictional-Defects),
                                                Order(Show-Cause),
                                                Frame(Footer) ]

Sequence(Untimely-NOA) = Left-To-Right [ Establish(NOA-Commencement),
                                         Establish(NOA-Filing-Date),
                                         Link(Furthermore),
                                         Establish(NOA-Due-Date),
                                         Link(Thus),
                                         Rule(Untimely-NOA) ]

Sequence(Finality-Defect) = Left-To-Right [ Establish(Order-Appealed-From),
                                            Link(Further),
                                            Establish(Non-Final-Order) ]

Sequence(Show-Cause) = Left-To-Right [ Link(Response-Due-Date),
                                       Order(Response-Due-Date),
                                       Order(Sanction) | Order(Partial-Sanction),
                                       Order(Non-Final-Order-Rationale) |
                                         Order(Untimely-NOA-Rationale) ]

Sequence(Appeals) = Left-To-Right [ Inform(Appeal),
                                    <main-appeal>,
                                    Link(However) ]

Sequence(Appeals) = Left-To-Right [ Inform(Appeal-Both),
                                    <main-appeal>,
                                    Link(And),
                                    <followup-appeal>,
                                    Link(However) ]

Sequence(Untimely-Main-Appeal) = Left-To-Right [ Inform(Untimely-Main-Appeal),
                                                 Link(As-To),
                                                 <main-appeal> ]
```

Appendix B

THE DISCOURSE STRUCTURE CONSTRUCTEDF OR SMYTHE

This appendix presents the discourse structure produced by the document planner by unifying the *Smythe* case facts with the document grammar for generating show-cause orders. The instantiated illocutionary and rhetorical structures are interleaved via the sequencing constraints. For purposes of presentation, the structure is depicted as a strict hierarchy rooted at (Establish(Preqs-for-Dismissal)). The document drafter interprets this structure to create the *Smythe* show-cause order shown in Figure 4.

```
Establish(Preqs-for-Dismissal) -->
     Frame(Header) -->
          "Colorado Court of Appeals                  Order"
          <case-number> = "No. 95CA0437"
          <order-number> = "Tr. Ct. No. 91CV051"
          Frame(Referent) -->
               Frame(Domestic) -->
                    "In re the Marriage of"
          <appellant> = "Herbert W. Smythe"
          "                       Appellant"
          "and"
          <appellee> = "Catherine Smythe"
          "                       Appellee"
          "To: "
          <appellant> = "Herbert W.  Smythe"
          Frame(Attorneys) -->
               Frame(Pro-Se) -->
                    "pro se appellant"
     Link(Preamble) -->
          "From the notice of appeal filed by appellant and the
                register of actions submitted by the clerk of the
                district court, it appears that"
     Establish(Jurisdictional-Defect) -->
          Establish(Appeals) -->
               Find(No-Main-Appeal) --> e
          Establish(Untimely-Defect) -->
               Establish(Untimely-NOA) -->
                    Establish(NOA-Commencement-Date) -->
                         Find(Judgment) -->
                              Find(One-Appeal-Only) -->
                              Inform(One-Appeal-Only) -->
                                   "judgment was entered"
                    Find(NOA-Commencement) -->
                         Find(Mailed) -->
                              <judgment-date> = "September 20, 1995"
                              Inform(Mailed) -->
                                   "and mailed to counsel of record on"
                              <mailing-date> = "September 22, 1995"
               Establish(NOA-Filing-Date) -->
                    Find(NOA-Filing-Date) -->
                         Link(And) -->
                              "and"
                         Inform(Appeal-Filed) -->
                              "the notice of appeal was filed on"
                         <filing-date> = "November 7, 1995"
               Establish(NOA-Due-Date) -->
                    Find(NOA-Due-Date) -->
                         Link(Furthermore) -->
                              "Furthermore, it appears that"
                         Inform(Appeal-Due) -->
                              "the notice of appeal was due"
                         Compute-Due-Date(<judgment-date>, <mailing-date>)
                              = "November 5, 1995"
               Rule(Untimely-NOA) -->
                    Rule(No-Main-Appeal) -->
                         Link(Thus) -->
                              "Thus, it appears that"
                         Inform(Untimely-Appeal) -->
                              "the notice of appeal was not timely filed"
                    Rule(Authority) -->
                         <authority> = "See C.A.R. 4(a)."
Order(Show-Cause) -->
     Link(Show-Cause) -->
          "IT IS THEREFORE ORDERED that the appellant shall
                show cause, if any there is, in writing on or before"
     Order(Response-Due-Date) -->
          Compute-Response-Date(<judgment-date>, <mailing-date>)
               = "December 18, 1995"
     Order(Sanction) -->
          Order(Final-Sanction) -->
               Inform(Final-Sanction) -->
                    "why this appeal should not be dismissed with prejudice"
     Order(Untimely-NOA-Rationale) -->
          Inform(Untimely-NOA-Rationale) -->
               "for failure to file a timely notice of appeal"
Frame(Footer) -->
     "                       BY THE COURT"
     "Date: "
     <order-date> = "December 4, 1995"
     "Copies to:  Counsel of Record"
```

Acknowledgments

The authors wish to thank: Anne Gill, staff attorney of the Colorado Court of Appeals, for assisting us in understanding jurisdictional screening procedures; Michael Elhadad of Ben Gurion University for his generous assistance with FUF;

Bruce Porter of the University of Texas at Austin for sharing the computing resources of his laboratory; Trevor Bench-Capon of the University of Liverpool, Giovanni Sartor of The Queen's University of Belfast, and Gary Stelling and Stuart Towns of North Carolina State University for insightful comments on earlier drafts of this manuscript; and Stuart Towns for assistance in creating the figures. This research is supported in part by a grant from the National Center for Automated Information Research and by NSF Faculty Early Career Development Grant IRI-9502152.

References

Allen, J. (1987) *Natural Language Understanding*. Benjamin/Cummings, Menlo Park, California.

Appelt, D.E. (1985) 'Planning English referring expressions', *Artificial Intelligence* 26, 1–33.

Austin, J. (1962) *How to Do Things with Words*. Oxford U. Press, New York.

Bench-Capon, T. and Staniford, G. (1995) 'PLAID – proactive legal assistance', In *Proceedings of the Fifth International Conference on Artificial Intelligence and Law*, pp. 81–88.

Branting, L. and Lester, J.C. (1996a) 'A framework for self-explaining legal documents', In *Proceedings of the Ninth International Conference on Legal Knowledge-Based Systems (JURIX-96)*, Tilburg University, the Netherlands, pp. 77–90.

Branting, L. and Lester, J.C. (1996b) 'Justification structures for document reuse', In *Proceedings of the Third European Workshop on Case-Based Reasoning (EWCR-96)*, Lausanne, Switzerland, pp. 76–90.

Branting, L.K. (1993a) 'An issue-oriented approach to judicial document assembly', In *Proceedings of the Fourth International Conference on Artificial Intelligence and Law*, Amsterdam, The Netherlands. ACM Press, pp. 228–235.

Branting, L.K. (1993b) 'A reduction-graph model of ratio decidendi', In *Proceedings of the Fourth International Conference on Artificial Intelligence and Law*, Amsterdam, The Netherlands. ACM Press, pp. 40–49.

Cawsey, A. (1992) *Explanation and Interaction: The Computer Generation of Explanatory Dialogues*. MIT Press.

Cohen, P.R. and Perrault, C.R. (1979) 'Elements of a plan-based theory of speech acts', *Cognitive Science* 3, 177–212.

Daskalopulu, A. and Sergot, M. (1995) 'A constraint-driven system for contract assembly', In *Proceedings of the Fifth International Conference on Artificial Intelligence and Law*, pp. 62–70.

DiMarco, C., Hirst, G., and Wanner, L. (1995) 'HealthDoc: Customizing patient information and health education by medical condition and personal characteristics', In *Working Notes of the Workshop on Artificial Intelligence in Patient Education*.

Elhadad, M. (1991) 'FUF: The universal unifier user manual version 5.0. Technical Report CUCS-038-91', Department of Computer Science, Columbia University.

Elhadad, M. (1992) *Using Argumentation to Control Lexical Choice: A Functional Unification Implementation*. PhD thesis, Columbia University.

Gordon, T. (1989) 'A theory construction approach to legal document assembly', In *Pre-Proceedings of the Third International Conference on Logic, Informatics, and Law*, Florence, pp. 485–498.

Grice, H. (1975) 'Logic and conversation', In Cole, P. and Morgan, J. (eds.), *Syntax and Semantics 2: Speech Acts*, Academic Press, New York, N.Y., pp. 41–58.

Grosz, B. and Sidner, C. (1986a) 'Attention, intention, and the structure of discourse', *Computational Linguistics* 12(3).

Grosz, B.J. and Sidner, C.L. (1986b) 'Attention, intentions, and the structure of discourse', *Computational Linguistics* 12(3): 175–204.

Hobbs, J. (1979) ‘Coherence and co-reference’, *Cognitive Science* 3(1), 67–82.

Hovy, E.H. (1990) ‘Pragmatics and natural language generation’, *Artificial Intelligence* 43, 153–197.

Hovy, E.H. (1993) ‘Automated discourse generation using discourse structure relations’, *Artificial Intelligence* 63, 341–385.

Johnson, W.L. (1994) ‘Dynamic (re)generation of software documentation’, In *Proceedings of the Fourth Systems Reengineering Technology Workshop*, pp. 57–66.

Kay, M. (1979) ‘Functional grammar’, In *Proceedings of the Berkeley Linguistic Society*.

Kittredge, R., Korelsky, T., and Rambow, O. (1991) ‘On the need for domain communication knowledge’, *Computational Intelligence* 7(4), 305–314.

Korelsky, T., McCullough, D., and Rambow, O. (1993) ‘Knowledge requirements for the automatic generation of porject management reports’, In *Proceedings of the Eigth Knowledge-Engineering Conference*. IEEE Computer Society Press.

Lauritsen, M. (1992) ‘Technology report: Building legal practice systems with today’s commericial authoring tools’, *Law and Artificial Intelligence* 1(1).

Lauritsen, M. (1993) ‘Knowing documents’, In *Fourth International Conference on Artificial Intelligence and Law*, Amsterdam. ACM Press, pp. 185–191.

Lester, J.C. and Porter, B.W. (1996) ‘Scaling up explanation generation: Large-scale knowledge bases and empirical studies’, In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pp. 416–423.

Lester, J.C. and Porter, B.W. (1997) ‘Developing and empirically evaluating robust explanation generators: The KNIGHT experiments’, *Computational Linguistics* 23(1), 65–101.

Mann, W.C. and Thompson, S.A. (1987) ‘Rhetorical structure theory: A theory of text organization’, Technical Report ISI/RS-87-190, USC/Information Sciences Institute, Marina del Rey, CA.

Marshall, C. (1989) ‘Representing the structure of a legal argument’, In *Proceedings of the Second International Conference on Artificial Intelligence and Law*, Vancouver, B.C., pp. 121–127.

Maybury, M.T. (1992) ‘Communicative acts for explanation generation’, *International Journal of Man-Machine Studies* 37(2), 135–172.

McKeown, K. (1982) *Generating Natural Language Text in Response to Questions about Database Structure*. PhD thesis, University of Pennsylvania.

McKeown, K. (1985) *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*. Cambridge University Press.

McKeown, K., Robin, J., and Kukick, K. (1995) ‘Generating concise natural language summaries’, *Information Processing and Management*. Special Issue on Summarization.

Mittal, V. (1993) *Generating Natural Language Descriptions with Integrated Text and Examples*. PhD thesis, University of Southern California.

Moore, J.D. (1995) *Participating in Explanatory Dialogues*. MIT Press.

Moore, J.D. and Paris, C.L. (1993) ‘Planning text for advisory dialogues: Capturing intentional and rhetorical information’, *Computational Linguistics* 19(4), 651–694.

Moore, J.D. and Swartout, W.R. (1991) ‘A reactive approach to explanation: Taking the user’s feedback into account’, In Paris, C.L., Swartout, W.R., and Mann, W.C. (eds.), *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, Kluwer Academic Publishers, Boston, pp. 3–48.

Paris, C. (1988) ‘Tailoring object descriptions to a user’s level of expertise’, *Computational Linguistics* 14(3), 64–78.

Pethe, V.P., Rippey, C.P., and Kale, L.V. (1989) ‘A specialized expert system for judicial decision support’, In *Proceedings of the Second International Conference on Artificial Intelligence and Law*, Vancouver, B.C., pp. 190–194.

Searle, J. (1969) *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, Cambridge.

Snellenburg, A.C. (1989) ‘New approaches to reducing court delay and congestion’, *State Court Journal* 13(3).

Spirgel-Sinclair, S. (1988) ‘The DHSS retirement pension forecase and advice system’, In Duffin, P. (ed.), *KBS in Government 88*, Blenheim On Line, Pinner, pp. 89–106.

Suthers, D.D. (1991) ‘A task-appropriate hybrid architecture for explanation’, *Computational Intelligence* 7(4), 315–333.

Suthers, D.D. (1993) *An Analysis of Explanation and Its Implications for the Design of Explanation Planners*. PhD thesis, University of Massachusetts.

Toulmin, S.E. (1958) *The Uses of Argument*. Cambridge University Press.

Zeleznikow, J. and Stranieri, A. (1995) ‘The Split-Up system: Integrating neural networks and rule-based reasoning in the legal domain’, In *Proceedings of the Fifth International Conference on Artificial Intelligence and Law*, pp. 185–194.