Automated Drafting of Self-Explaining Documents

L. Karl Branting
Department of Computer Science
University of Wyoming
Box 3682
Laramie, WY 82071
karl@index.uwyo.edu

James C. Lester **Charles B. Callaway**
Department of Computer Science
North Carolina State University
Box 8206
Raleigh, NC 27606
lester@adm.csc.ncsu.edu
cbcallaw@eos.ncsu.edu

Abstract

The capacity for self-explanation can make computer-drafted documents more credible, assist in the retrieval and adaptation of archival documents, and permit comparison of documents at a deep level. We propose a knowledge-based model of documents that makes explicit the underlying goals that documents are intended to achieve and the stylistic conventions to which they must conform. These goals and conventions are expressed in a dual justification structure that represents the illocutionary and rhetorical dependencies underlying documents.

After demonstrating how a document grammar derived from dual justification structures can be used to automate document drafting, we show how documents can exploit dual justification structures to "explain themselves" by answering queries about (1) the purposes for inclusion of text in the document and (2) the justification for propositions expressed in the text. This self-explanation framework has been implemented in the DOCU-PLANNER, a prototype document generation system that produces "queryable" documents.

1 Introduction

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 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.

Explicit representation of these goals and stylistic conventions can benefit automated creation and maintenance of documents in several ways. First, users are likely to give greater credence to computer-drafted documents if the reasons underlying the selection and arrangement of text elements are accessible to the user. Second, explicit representation of document goals and stylistic conventions aids revision and maintenance of legacy documents by assisting document indexing, matching, and adaptation. Libraries of documents, such as briefs and contracts, constitute an institutional memory for many organizations such as law firms, corporations, and courts. Typically, the most relevant archival document for a given task is the document that shares the greatest number of goals with the current task. Thus, explicit representation of document goals facilitates indexing. Adapting an archival document requires understanding the goals and stylistic conventions underlying the document. If these are made explicit and recorded when the document is created, they can be accessed to assist in manual or automatic adaptation of the document for use in a new situation. Finally, explicit representation of document goals permits comparison between different versions of a document, such as revisions or alternative drafts proposed during negotiations, at the level of goals rather than at the word level.

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 and in indexing, explaining, and adapting existing documents. The goals that a document achieves are expressed as a dependency tree whose nodes consist of *illocutionary* operators, and the stylistic conventions of the genre are expressed in a dependency tree whose nodes consist of *rhetorical* operators.[1] Together, these two dependency trees constitute the *dual justification structure* of a document. The illocutionary and rhetorical operators capable of describing the dual justification structures of a set of documents constitutes a *document grammar* for the documents. We say that a document containing a given discourse is *self-explaining* if it contains an explicit representation of the illocutionary and rhetorical structure underlying the discourse.

We have previously argued that representation of the dual justification structures of documents can facilitate (1) document drafting (2) retrieval, interpretation, and adaptation of previous documents, and (3) maintenance of multi-

[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 [All87] for a more detailed discussion.

generation documents [BL96b, BL96a, BLC97]. In this paper we focus on the use of dual justification structures for explanation. We show that the dual justification structure approach supports explanations about both why a particular clause was included in the document and how the proposition expressed in the clause was justified.

Section 2 describes a representative class of legal documents, appellate jurisdictional show-cause orders, illustrates how these documents can be represented in terms of dual justification structures, and describes how a document grammar derived from these dual justification structures can be used in the drafting of new show-cause orders. Section 3 shows how the dual justification structures in a document created from a document grammar can be used to answer questions about why a particular clause was included in the document and how the proposition expressed in the clause was justified. Related work is discussed in Section 4, and Section 5 sets forth the scope of this approach and future research.

2 A Document Grammar for Performative Legal Documents

2.1 Jurisdictional Show-Cause Orders

Attorneys, judges, and other legal personnel draft documents of a wide range of complexity. The most complex documents are probably appellate opinions. The complexity and individuality of appellate opinions makes automated assistance for such documents far beyond the scope of current technology. At the opposite extreme of complexity are documents consisting almost entirely of boilerplate, such as motions for extensions of time or notices of hearing dates. An intermediate level of complexity is occupied by documents that have sufficient stylistic and substantive consistency to lend themselves to formal characterization but are too complex to be amenable to simple template-based document-drafting techniques. Jurisdictional 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.

Jurisdictional show-cause orders are issued by an appellate court in response to an apparent jurisdictional defect in a case brought for appeal. The purpose of show-cause orders is to establish the prerequisites for dismissal of the appeal by (1) establishing an apparent defect and (2) ordering the appellant to rebut the apparent defect within a fixed interval or suffer a sanction (dismissal with or without prejudice, that is, with or without the option of bringing the appeal again at some later time).

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 early as possible in the appeals process.

The Colorado Court of Appeals, where one of the authors, Karl Branting, worked for several years as a staff attorney, 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.

```
---------------------------------------------------
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.

Figure 1 shows a typical show-cause order. This order establishes 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.

2.2 Illocutionary and Rhetorical Structure of Legal 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 establish or alter legal relationships relevant to some controversy before the court. Thus, the purpose of a show-cause order is not merely to inform an appellant of a potential problem, but to satisfy the procedural prerequisites for a change in the appellant's legal status.

There are generally three requirements that a performative judicial document, such as an order or decision, must satisfy to achieve the goal of establishing 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.[2]

[2] The findings of facts are analogous to the data in Toulmin's

For example, the show-cause order set forth in Figure 1 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 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 tree differs from a conventional goal tree in two ways. First, 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, the illocutionary goal of establishing the date when the notice of appeal was filed requires including in the document text that makes a finding of the filing date.

A second difference between an illocutionary tree and a conventional goal tree is that all internal nodes of an illocutionary tree (other than find, rule, and order) consist of the illocutionary goal of establishing a proposition. This is in contrast to a conventional goal tree in which the goal is simply the proposition itself. For example, in a conventional goal tree, the goal untimely(Notice-of-appeal) might have as its subgoal filing-date(Notice-of-appeal, ?Some-date). In an illocutionary tree, the goals are not these propositions *per se*, but are instead the illocutionary goals of establishing these propositions. Thus, the internal nodes in the illocutionary tree shown in Figure 2 (other than find, rule, and order) are to establish a proposition.

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 goals, in turn, have as subgoals: 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 subgoals, 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 goal structures constitute the *dual justification structure* of a document.

In summary, the dual justification structure of a document consists of illocutionary and rhetorical structures 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 illocutionary and rhetorical structures are composed, respectively, of illocutionary and rhetorical operators that ground out in the text of the document.

[Tou58] 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.

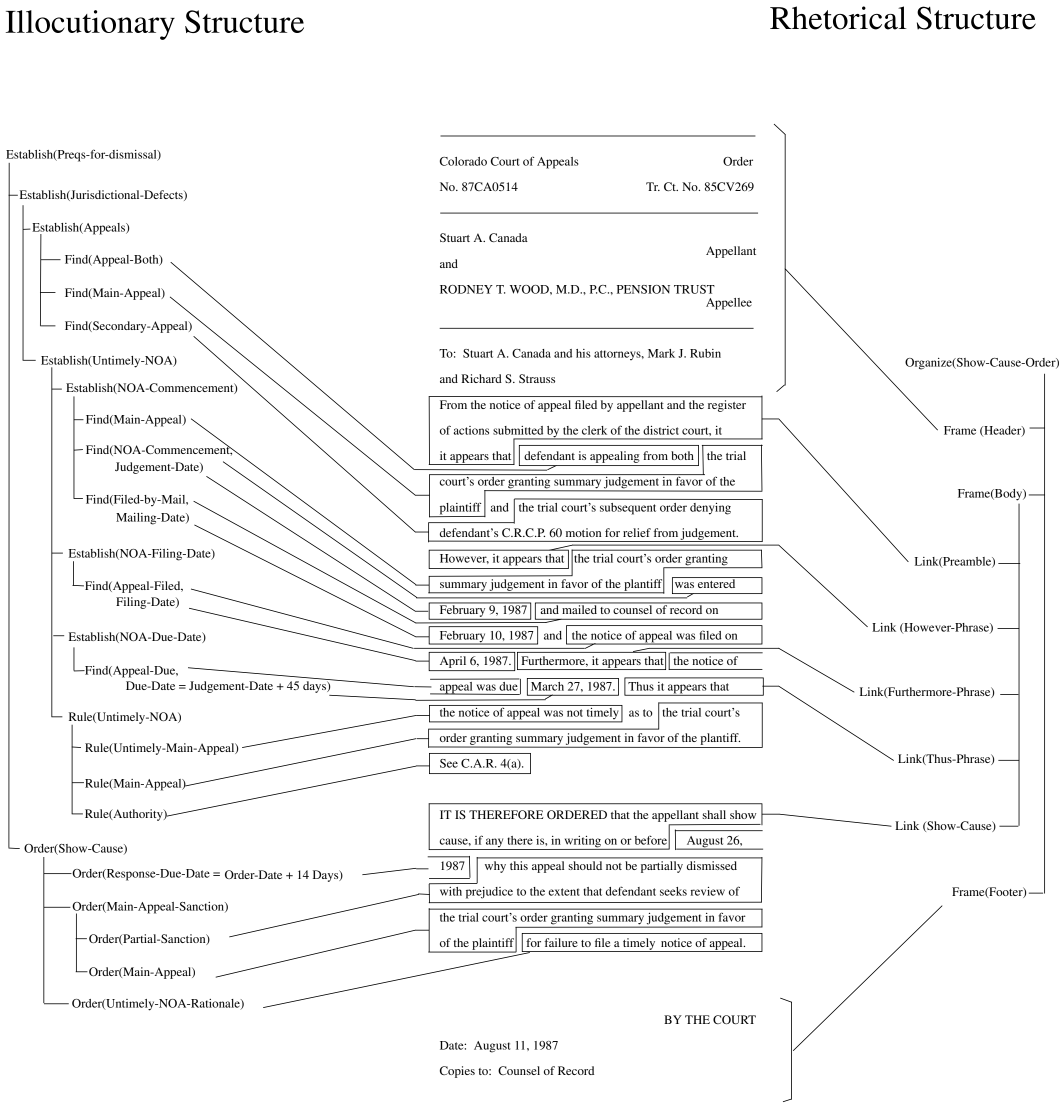

Figure 2: The illocutionary and rhetorical structure of *Canada*

2.3 Using Document Grammars for Drafting

The illocutionary and rhetorical operators necessary to construct the justification 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 dual justification 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.

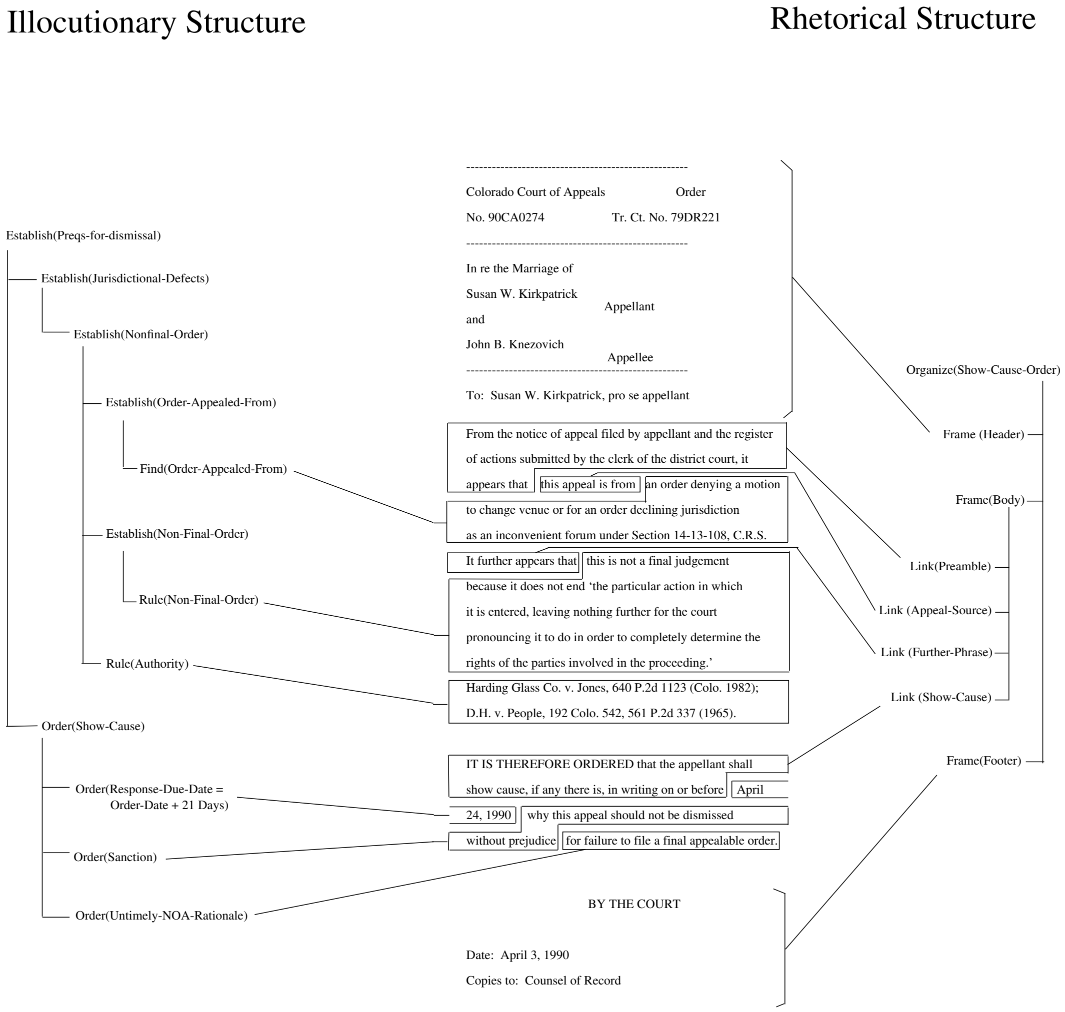

Figure 3: The illocutionary and rhetorical structure of *Kirkpatrick*

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. These facts might be gathered through electronic forms with entries corresponding to possible case values, through a conventional backward-chaining jurisdictional rule-based system, or through a mixed-initiative approach under which the user can directly provide case facts, ask the system for examples, counter-examples or advice on answering questions, or invoke an inference engine to help infer facts. Suppose that these facts, together with the docket numbers on appeal and at trial, attorneys' names, *etc.*, were 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.

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 dual justification structures is described in [BLC97].

The next section illustrates how the dual justification structures formed when a document is produced from a doc-

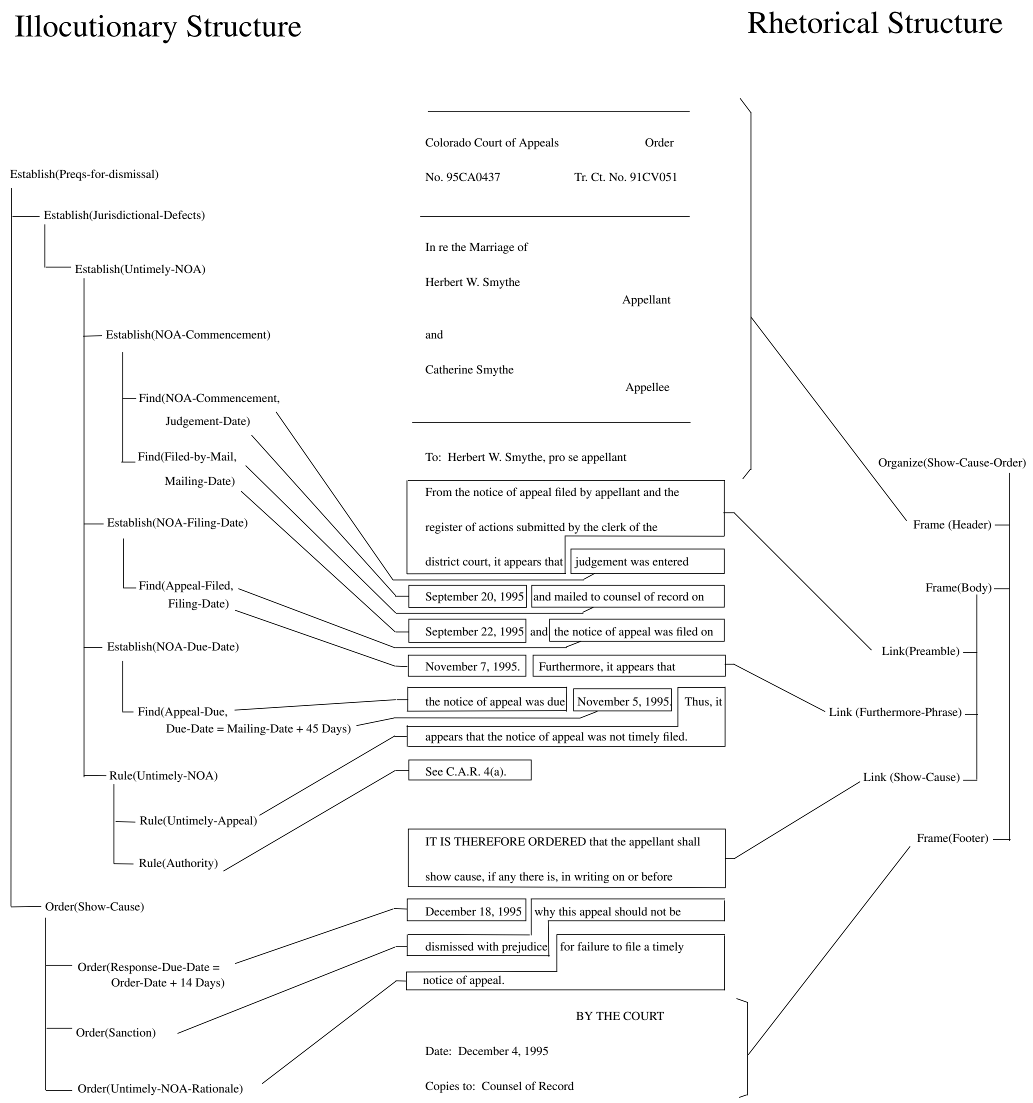

Figure 4: The illocutionary and rhetorical structure of *Smythe*

ument grammar can be used to provide two distinct types of explanations.

3 Explanation Using Dual Justification Structures

3.1 Self-Explaining Documents

A self-explaining document is one that contains an explicit representation of the illocutionary and rhetorical structure underlying the document's text and is capable of providing explanations by exploiting this structure. In this section, we demonstrate how the dual-justification structures created when a document is drafted using the document grammar described in the previous section can be used to provide answers to important questions about the composition of the document.

Document explanation is the task of responding to users' questions about specified segments in documents. More formally, given a document D, a specific segment S of D, and a query Q about S, the task of the explanation system is to produce an explanation in response to Q that justifies S in D. Q may be one of two types:

- *Rationale Justification:* How is the conclusion expressed by this text justified?
- *Inclusion Justification:* Why is this text included in the document?

To construct appropriate explanations, self-explaining documents must attend to the type of query posed. Rationale-justification queries are appropriate for document segments that express a conclusion. Providing explanations to rationale justification queries requires referring to the operators that were used to establish the specified conclusion. Inclusion justification queries can be answered with two types of responses. First, text might have been included in the document because of rhetorical constraints, *i.e.,* because the text is required for documents of the given genre. Alternatively, the text might have been included to satisfy illocutionary goals, *e.g.,* to find, rule, or order.

Users may pose queries in two contexts. First, they may issue *initial* probes of a particular segment, in response to which the document system will provide an explanation. Second, they may issue *secondary* probes to obtain explanations of previously issued explanations. Secondary probes are follow-up questions that enable users to inquire in depth about inclusion or rational justifications.

The self-explaining document framework exploits the dual justification structure underlying each document to provide explanations. Associated with each operator in a document grammar are one or more *explanatory annotations* stating the relationship between the super-goal and sub-goals of the operator. Illocutionary operators may house either a rationale justification annotation and/or an inclusion justification; rhetorical operators house only inclusion justification annotations.

As a document is generated, the document planner embeds the relevant explanatory annotations into the resulting dual justification structure (Figure 5). As the planner constructs the illocutionary structure, it incorporates explanatory annotations from the instantiated illocutionary operators into the document's illocutionary structure. As it constructs the rhetorical structure, it incorporates explanatory annotations from the instantiated rhetorical operators into the document's rhetorical structure.

Documents explain themselves according to the following algorithm. When a user specifies a particular segment of the text, the explanation system first identifies instantiated operators in the dual justification structure that are associated with the selected segment. If the user poses a rationale justification query, *i.e.*, "How is the conclusion expressed by this text justified?" the system ascends the illocutionary structure to obtain the rationale justification annotation associated with the bottom-most illocutionary operators that were used to generate the segment specified by the user. If the user poses an inclusion justification query, *i.e.*, "Why is this text included in the document?" the system inspects the dual justification structure. If the specified segment was produced by the sub-goaling of illocutionary operators, it ascends the illocutionary structure and locates the inclusion justification annotation associated with bottom-most illocutionary operators in the tree. Alternatively, if the specified segment was produced by the sub-goaling of rhetorical operators, it ascends the rhetorical structure and locates the rational justification annotation associated with the bottom-most rhetorical operators in the tree. Finally, it marks the operator with the relevant explanatory annotation and presents the explanation.

If the user poses a secondary query to request a follow-up explanation, the system begins its ascent at the marked operator. It climbs to the lowest explanatory annotation that (a) inhabits a super-goal of the goal associated with the marker and (b) is of the same type (rationale or inclusion) as the marker. It then resets the marker and presents the explanation to the user. Follow-ups continue until either the ascent reaches the operator associated with the top-most goal or the user requests no additional explanations.

3.2 Implementation and Example

The self-explaining document framework has been implemented in the DOCU-PLANNER, a prototype document planning system automatically creates show-cause orders that identify apparent defects and issue orders to appellants. The DOCU-PLANNER builds on a large body of work in computational linguistics on multi-sentential discourse generation.[3]

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 dual justification structure in which the operators defining the illocutionary and rhetorical structures are fully instantiated, linked together through variable bindings, and accompanied by explanatory annotations. 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 dual justification 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 justification structure) in the concatenated text. Finally, it interprets the resulting linear structure, thereby creating the completed document. Once the document has been generated, the user may ask it to "explain itself."

To illustrate, suppose that the *Kirkpatrick* show-cause order shown in Figure 3 has either (1) just been drafted from the document grammar described above as applied to a set of facts determined by a staff attorney, or (2) has been retrieved on the basis of the similarity of its illocutionary or rhetorical structure to the goals to be satisfied in a new case. Now suppose that the user selects the header of the *Kirkpatrick* case.[4] In this case, only an inclusion justification can be requested. The response of the system to this question is: "A show-cause order begins with a header that sets forth the parties, docket number on appeal, docket number in the forum below, and the party to whom the show cause order is directed," which is the annotation associated with the header operator.

If the user now selects the segment "an order denying a motion to change venue or for an order declining jurisdiction as an inconvenient under Section 14-13-108, C.R.S." he or she may then pose either a rational justification or an inclusion justification question. Based on the explanatory annotation associated with the illocutionary structure of which this text is a leaf node, the answer to an inclusion query, *i.e.*, "Why is this text included in the document?" is

> To establish that the order is nonfinal, the document must (1) make a finding as to the type of order that the appellant is appealing from and (2) rule that an order of this type is not final. This text makes a finding as to the type of order that the appellant is appealing from.

If the user poses a follow-up question asking "why" again, the system ascends the goal tree and produces the response, "The document is attempting to establish that there is a jurisdictional defect by finding that the appeal is untimely, nonfinal, or has a subject defect." If the user asks "why" yet again, the explanation is "To establish the prerequisites for dismissal a show-cause order must (1) establish the existence of a jurisdictional defect and (2) order a sanction unless a response is made within a time limit." In each case, the responses are obtained by climbing the illocutionary goal tree and extracting the associated inclusion justification annotations.

[3] The planner is implemented in a *unification-based* formalism [Elh91], and the explanation system is implemented in Harlequin Lisp on a DEC Alpha.

[4] The current prototype provides only I/O with the Lisp interpreter; a WWW-based GUI will be added soon.

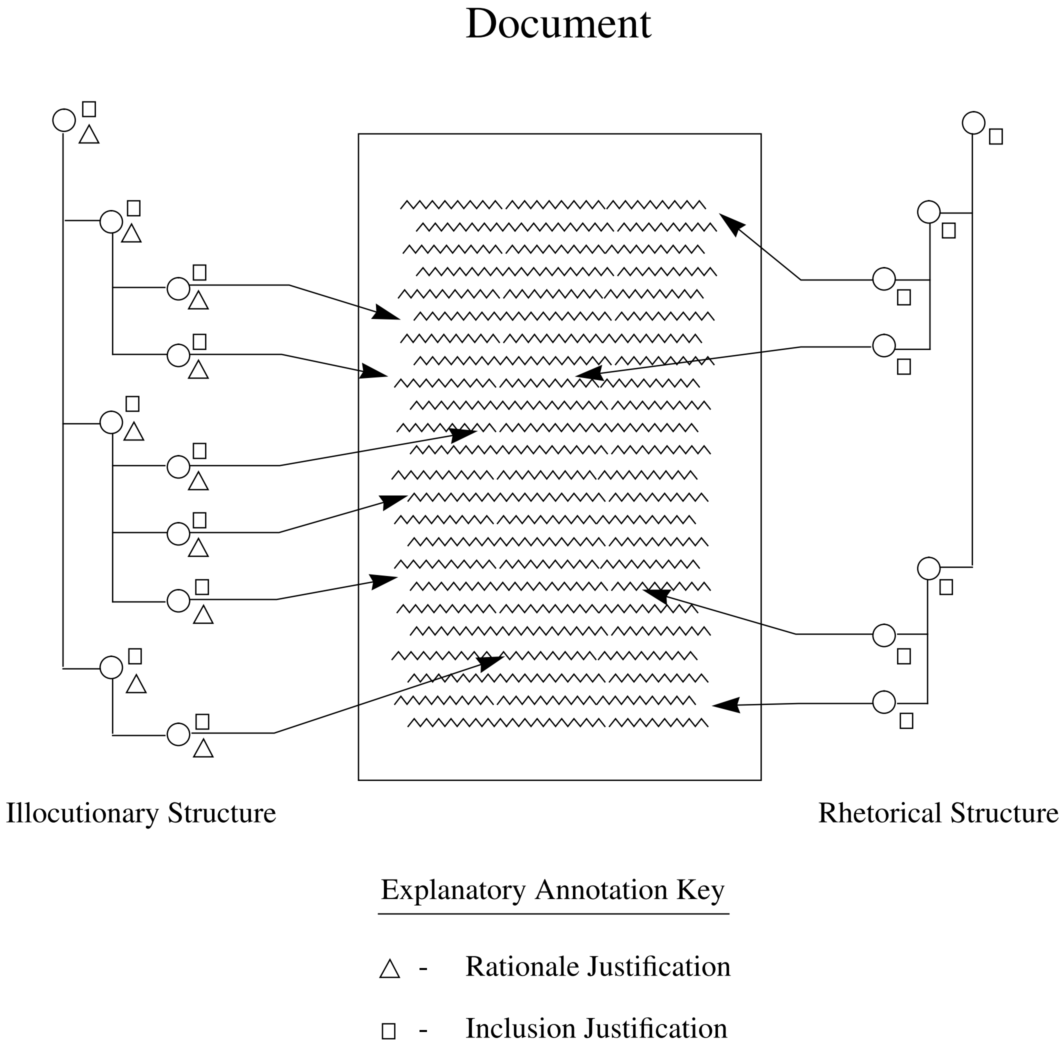

Figure 5: Explanatory annotations embedded in dual structure

Now suppose the user requests a rationale justification for the same segment of the document, *i.e.*, he or she asks, " How is the conclusion expressed by this text justified?" Because the nature of the order appeal was simply provided in the original draft in the facts of the *Kirkpatrick* case, the resulting explanation is simply

> That the order being appealed is a motion to change venue or for an order declining jurisdiction as an inconvenient under Section 14-13-108, C.R.S. was stated in the given fact.

Now suppose the user selects the segment, "this is not a final judgment because ...". He or she can pose either an inclusion query or a rationale query. Selecting an inclusion query produces the following explanation by climbing the illocutionary structure:

> To establish that the order is nonfinal, the document must (1) make a finding as to the type of order that the appellant is appealing from and (2) rule that an order of this type is not final. This text makes a ruling that the judgment being appealed is non-final.

If the user asks "why" again, the same explanations will be produced as were produced in response to repeated "why" questions concerning the text "this appeal is from an order denying a motion to change venue ... " because both of these text segments share common illocutionary supergoals. Alternatively, if the user requests a rationale justification, the explanation identifies the warrant for the rule under which the conclusion is justified

> Under C.A.R. 4(a), an order that does not resolve all claims as to all parties is not final. An order denying a motion to change venue does not resolve all claims as to all parties. Therefore, the order being appealed from is not final.

Finally, suppose the user selects the segment "...why this appeal should not be dismissed without prejudice" and requests an inclusion justification. The system explains that, "A show cause order must order a sanction and identify a response due date and a rationale. This text orders the sanction." When the user asks "why" again, the response is ""To make an appropriate show-cause order the document must (1) order a sanction unless a response is made within a time limit and (2) provide a rationale for doing so." Now suppose the user completes his or her interrogation of the document by requesting a rationale justification for the same segment, to which the system responds, "The sanction for lack of jurisdiction because of nonfinality is dismissal without prejudice, because the judgment may become final at some later time."

4 Related Work

The explanation community has extensively studied the process of planning and realizing text given a set of discourse specifications. Over the past decade, their research on discourse planning [McK85, Par88, Hov93, Caw92, Moo95, LP97] 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 multisentential text, a key capability of self-explaining documents.

Our approach to self-explaining documents draws on four different lines of research: explanation generation, discourse structure analysis, the theory of argumentation, 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.*, [GS86, Hob79]. The formalization of inter-sentential discourse relations is a key requirement for the development of self-explaining documents.

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 [Aus62], speech act theory addresses the illocutionary content of discourse, *i.e.*, the goals that a speaker intends to accomplish through that discourse [Gri75, Sea69].

The theory of argumentation addresses texts intended to persuade, establish, or prove. For example, Toulmin [Tou58] analyzed argumentative texts in terms of the concepts of warrant, ground, conclusion, backing, and qualification. This model has been widely applied to the analysis [Mar89, ZS95] and creation [BCS95] 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 [Bra93].

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 [Lau92]. Some progress has been made in exploiting explicit representations of the relationship between generic documents and document instances and of constraints among document components [DS95]. 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 [Gor89, Lau93].

The applied computational linguistics community has addressed another form of text production from an underlying domain structure: automated report generation. 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 [KKR91]. 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 [KMR93, Joh94, MRK95] and on producing customized patient information reports for medical applications [DHW95].

5 Discussion and Future Work

In this paper we have presented a model of the illocutionary and rhetorical structures underlying a representative legal document—jurisdictional show-cause orders. We have shown how these structures can be used to form a document grammar that can be used to (1) generate new documents using a unification-based procedure, and (2) enable documents to "explain themselves" by answering queries about the purposes for inclusion of text in the document and the justification for propositions expressed in the text. We have argued that the capacity for self-explanation can make computer-drafted documents more credible, assist in the retrieval and adaptation of archival documents, and permit comparison of documents at a deep level. This approach has been implemented in the DOCU-PLANNER, a prototype document generation system that produces "queryable" documents. By embedding explanatory annotations from document planning operators into the illocutionary and rhetorical structures of a document, and then exploiting these annotations to respond to initial queries and follow-up inquiries, the DOCU-PLANNER provides users with the ability to pose multiple types of questions about particular segments of the document.

Having demonstrated the feasibility of the self-explaining document approach through the construction of a prototype grammar for show-cause orders, our current plans are to address the knowledge-acquisition issues in document grammar construction by building a larger scale implementation. We believe that the self-explaining grammar framework has potential applicability for both tutorial applications, such as law-school courses involving document drafting, and practical applications for large institutions with multi-generation archival documents. We will be exploring these in our future work.

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; and Stuart Towns of North Carolina State University 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

[All87] J. Allen. *Natural Language Understanding.* Benjamin/Cummings, Menlo Park, California, 1987.

[Aus62] J. Austin. *How to do things with words.* Oxford U. Press, New York, 1962.

[BCS95] T. Bench-Capon and G. Staniford. PLAID - proactive legal assistance. In *Proceedings of the Fifth International Conference on Artificial Intelligence and Law*, pages 81–88, 1995.

[BL96a] L.K. Branting and J. C. Lester. A framework for self-explaining legal documents. In *Proceedings of the Ninth International Conference on Legal Knowledge-Based Systems (JURIX-96)*, pages 77–90, Tilburg University, the Netherlands, December 1996.

[BL96b] L.K. Branting and J. C. Lester. Justification structures for document reuse. In *Proceedings of the Third European Workshop on Case-Based Reasoning (EWCR-96)*, pages 76–90, Lausanne, Switzerland, November 1996.

[BLC97] L. Branting, J. Lester, and C. Callaway. Automating judicial document drafting: A unification-based approach. *Artificial Intelligence and Law*, 1997. To appear.

[Bra93] L. K. Branting. An issue-oriented approach to judicial document assembly. In *Proceedings of the Fourth International Conference on Artificial Intelligence and Law*, pages 228–235, Amsterdam, The Netherlands, June 15–18, 1993. ACM Press.

[Caw92] A. Cawsey. *Explanation and Interaction: The Computer Generation of Explanatory Dialogues.* MIT Press, 1992.

[DHW95] Chrysanne DiMarco, Graeme Hirst, and Leo Wanner. 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*, 1995.

[DS95] A. Daskalopulu and M. Sergot. A constraint-driven system for contract assembly. In *Proceedings of the Fifth International Conference on Artificial Intelligence and Law*, pages 62–70, 1995.

[Elh91] M. Elhadad. FUF: The universal unifier user manual version 5.0. Technical Report CUCS-038-91, Department of Computer Science, Columbia University, 1991.

[Gor89] T. Gordon. A theory construction approach to legal document assembly. In *Pre-Proceedings of the Third International Conference on Logic, Informatics, and Law*, pages 485–498, Florence, 1989.

[Gri75] H. Grice. Logic and conversation. In P. Cole and J. Morgan, editors, *Syntax and Semantics 2: Speech Acts*, pages 41–58. Academic Press, New York, N.Y., 1975.

[GS86] B. Grosz and C. Sidner. Attention, intention, and the structure of discourse. *Computational Linguistics*, 12(3), 1986.

[Hob79] J. Hobbs. Coherence and co-reference. *Cognitive Science*, 3(1):67–82, 1979.

[Hov93] E. H. Hovy. Automated discourse generation using discourse structure relations. *Artificial Intelligence*, 63:341–385, 1993.

[Joh94] W. Lewis Johnson. Dynamic (re)generation of software documentation. In *Proceedings of the Fourth Systems Reengineering Technology Workshop*, pages 57–66, 1994.

[KKR91] R. Kittredge, T. Korelsky, and O. Rambow. On the need for domain communication knowledge. *Computational Intelligence*, 7(4):305–314, 1991.

[KMR93] T. Korelsky, D. McCullough, and O. Rambow. Knowledge requirements for the automatic generation of porject management reports. In *Proceedings of the Eigth Knowledge-Engineering Conference.* IEEE Computer Society Press, September 20–23 1993.

[Lau92] M. Lauritsen. Technology report: Building legal practice systems with today's commericial authoring tools. *Law and Artificial Intelligence*, 1(1), 1992.

[Lau93] M. Lauritsen. Knowing documents. In *Fourth International Conference on Artificial Intelligence and Law*, pages 185–191, Amsterdam, 1993. ACM Press.

[LP97] James C. Lester and Bruce W. Porter. Developing and empirically evaluating robust explanation generators: The KNIGHT experiments. *Computational Linguistics*, 23(1):65–101, 1997.

[Mar89] C. Marshall. Representing the structure of a legal argument. In *Proceedings of the Second International Conference on Artificial Intelligence and Law*, pages 121–127, Vancouver, B.C., June 13-16 1989.

[McK85] K. McKeown. *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text.* Cambridge University Press, 1985.

[Moo95] J. D. Moore. *Participating in Explanatory Dialogues.* MIT Press, 1995.

[MRK95] K. McKeown, J. Robin, and K. Kukick. Generating concise natural language summaries. *Information Processing and Management*, 1995. Special Issue on Summarization.

[Par88] C. Paris. Tailoring object descriptions to a user's level of expertise. *Computational Linguistics*, 14(3):64–78, September 1988.

[Sea69] J. Searle. *Speech Acts: An Essay in the Philosophy of Language.* Cambridge University Press, Cambridge, 1969.

[Tou58] S. E. Toulmin. *The Uses of Argument.* Cambridge University Press, 1958.

[ZS95] J. Zeleznikow and A. Stranieri. 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*, pages 185–194, 1995.