Justification Structures for Document Reuse*

L. Karl Branting[1], and James C. Lester[2]

[1] Department of Computer Science
University of Wyoming
Box 3682
Laramie, WY 82071
karl@index.uwyo.edu

[2] Department of Computer Science
North Carolina State University
Box 8206
Raleigh, NC 27606
lester@adm.csc.ncsu.edu

Abstract. Document drafting—an important problem-solving task of professionals in a wide variety of fields—typifies a design task requiring complex adaptation for case reuse. This paper proposes a framework for document reuse based on an explicit representation of the illocutionary and rhetorical structure underlying documents. Explicit representation of this structure facilitates (1) interpretation of previous documents by enabling them to "explain themselves," (2) construction of documents by enabling document drafters to issue goal-based specifications and rapidly retrieve documents with similar intentional structure, and (3) maintenance of multi-generation documents.

1 Introduction

Documents play an increasingly important role in all sectors of society. Legal documents precisely stipulate complex relationships between parties; government documents set forth regulatory requirements and procedures; and software documentation provides both specifications and usage recommendations. Because of the ubiquity of complex, formal documents, document interpretation and maintenance has become an issue of significant economic and legal import.

Complex documents are typically created by modifying previous documents. For example, adaptation and reuse of previous documents is an almost universal practice in U.S. law firms. Document reuse is beneficial because it promotes stylistic and substantive consistency and reduces drafting time.

However, document reuse requires access to the original intentions underlying the document, which may not be readily apparent from the document's

* This research is supported in part by grants from the National Center for Automated Information Research and by NSF Faculty Early Career Development Grant IRI-9502152.

surface text. For example, when both parties to a contract agree that modifications should be made, the assumptions behind the original contract must be reconstructed to determine the precise textual changes required. Thus, complex documents suffer from many of the same problems as legacy software: as the context for which a document was created changes, the document becomes outdated and requires revision. Problems of document maintenance are exacerbated by the frequent personnel changes that characterize large institutions, where many documents are created. Without access to a document's authors, the intent behind particular clauses may be lost, impeding interpretation and modification.

To address these problems, we propose a *self-explaining documents* framework. 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. Because self-explaining documents record intentional knowledge, they offer significant potential for the interpretation and maintenance of complex, multi-generation documents. In particular, they can explain why a particular clause was included, suggest how an existing document should be modified to apply to slightly different circumstances, and present arguments for the pros and cons of alternative clauses.

This paper presents a framework for the use of justification structures for document reuse. Section 2 describes three document management tasks that require self-explaining documents. Section 3 proposes a dual justification structure that combines illocutionary and rhetorical structures to represent document intent. Section 4 presents a computational architecture for interactive self-explaining document systems that assist users in constructing and querying documents, and Section 5 illustrates the envisioned behavior of the system when applied to the domain of will drafting. We are currently exploring the self-explaining document framework in the context of routine judicial orders. Section 6 describes the implementation plans for this work, and Section 7 outlines related work.

2 Applications of Self-Explaining Documents

Three tasks can be distinguished requiring knowledge of the illocutionary and rhetorical structure of texts: document drafting, document analysis, and document maintenance. Given a set of specifications—the formality of document specifications can range from formal, quasi-legal descriptions to sketchy, informal notes—the task of *document drafting* is to prepare a document that satisfies both the illocutionary goals of the specifications and the genre-specific rhetorical conventions. Document drafters typically operate in an iterative fashion, gradually refining the document until it satisfies the constraints noted above and, perhaps, meets with the approval of a client.

As stated above, drafting of complex documents is typically performed by modifying existing documents. The task of drafting by means of document reuse can be summarized as follows:

Given:

- A set of goals to be accomplished by the document to be drafted.
- A library of existing documents.

Do:

- Retrieval. Find the existing document(s) (or combination of document components) that best satisfy the current goals.
- Comparison. Display the differences, if any, between, the goals achieved by the retrieved document(s) and the current goals.
- Adaptation. Remove the portions of text whose only purpose is to satisfy goals that aren't present in the current situation (excision), and add text to satisfy any of the current goals not satisfied by the retrieved text (augmentation).

While document drafting is primarily a synthetic task, *document analysis* centers around interpretation. Here, the goal is to ascertain the intent of particular clauses, to identify documents that are similar, and to perform comparative analyses. In the course of conducting an analysis, users must be able to quickly locate relevant documents and precisely establish illocutionary analogies between statements in a given document and those in archival documents.

Document maintenance is concerned with revision. Because the overriding goal of maintenance operations is to revise an existing document to reflect new specifications, this task typically combines elements of both drafting and analysis. Existing versions of documents must be interpreted in light of new circumstances, and new sections or clauses are created to address the new context.

Because drafting, analysis, and maintenance are tightly interleaved in practice, an effective self-explaining document systems should offer a uniform communication mechanism that (1) employs a single representational vocabulary for specifying and querying documents, and (2) enables users to interleave specification and querying operations with ease.

A complete document management system[3] should enable users to pose four classes of queries:

- *Intentional Identification:* Why was a given segment of a document included?
- *Intentional Achievement:* What types of document segments in a given genre will achieve a given goal?
- *Intentional Exemplification:* What are some examples of archival document segments that achieve a given goal?
- *Intentional Comparison:* To what segments of archival documents is a given document segment most similar? What are the intentional similarities and differences between two documents (at the most specific abstraction level at which they differ)?

[3] We use "document management" as the super-ordinate concept for document drafting, document analysis, and document maintenance.

As users engage in document-building and document-querying dialogues, they should be able to shift effortlessly between discussions of abstract document features (*e.g.*, the predicates employed in illocutionary and rhetorical structures) and specific segments of particular documents (both documents under construction and archival documents).

The next section describes the representation requirements for the document drafting, analysis, and maintenance tasks.

3 Representational Requirements

The primary emphasis of most work in discourse analysis and generation has been on texts having communicative goals. However, documents are often intended to accomplish various other illocutionary goals, including eliciting information, persuading, memorializing events such as reciprocal communications, or accomplishing performative goals, such as creating or revoking legal, social, or institutional relationships. Self-explaining documents therefore require a rich vocabulary of illocutionary goals.

However, the illocutionary goal structure of a document is not *per se* sufficient to completely determine the selection and configuration of text. In general, the illocutionary goal structure leaves unspecified 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. Accordingly, self-explaining documents must include the rhetorical structure of documents as well in order to answer intentional identification and comparison queries.

The minimum representational requirements for self-explaining documents therefore include the following:

- A taxonomy of illocutionary goals sufficiently expressive to permit retrieval of documents, comparison of documents, and explanation of document components. The necessary granularity of the leaf nodes of this taxonomy depends on the requirements of the particular document genre and the pragmatics of the user's application.
- A taxonomy of rhetorical goals.
- A representation of templates at a level of granularity corresponding to the leaf nodes of the illocutionary goals. The substitutable elements of text templates should be tagged with a data-type so that entire documents, or document components, can be viewed either as uninstantiated templates or as fully instantiated texts.
- A set of link annotations, *e.g.*, annotations providing the legal authority under which a given legal goal is satisfied by a given set of performative subgoals, and annotations explaining why a given rhetorical goal is satisfied by a particular set of subgoals in a given document genre.

To illustrate, a simplified representation of the illocutionary structure of a will is shown in Figure 1. The root illocutionary goal is to make a bequest.

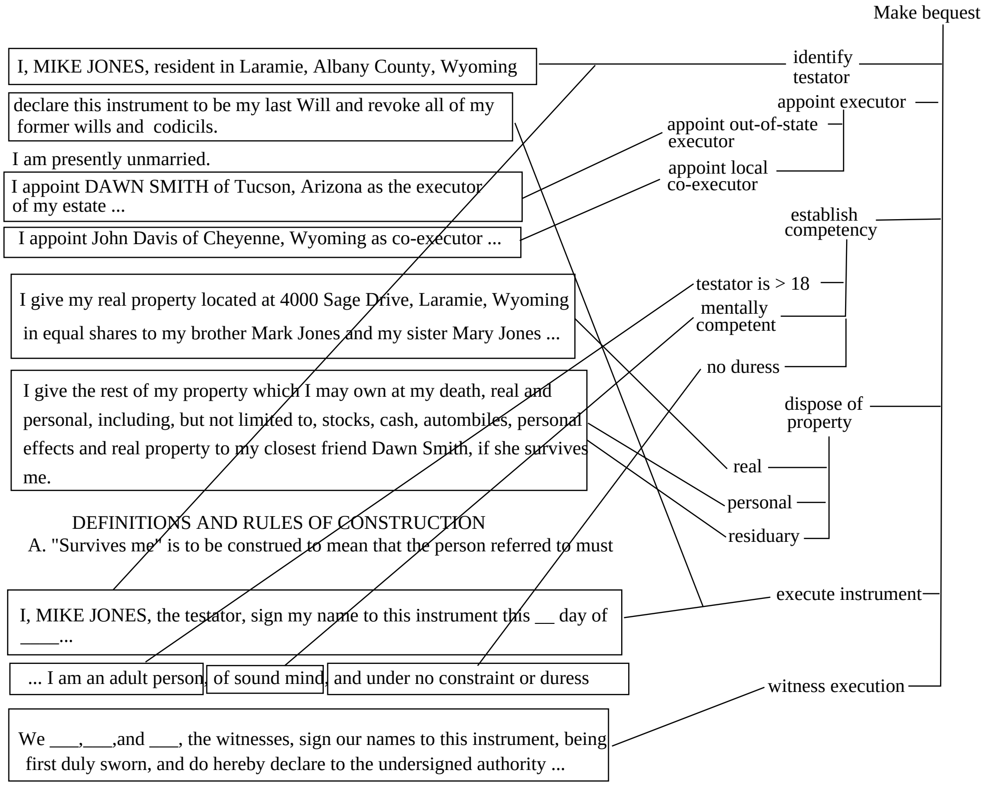

Fig. 1. A simplified representation of the illocutionary structure of a will.

Achieving this goal requires the operators displayed as children of the root: identify the testator, appoint an executor, etc. Each leaf illocutionary goal is connected to a text segment intended to achieve that goal. Not shown are the annotations explaining why (*i.e.*, citing the legal authority under which) making a bequest has the requirements shown.

A simplified representation of the rhetorical structure of the will is shown in Figure 2. Unlike the illocutionary structure, the rhetorical structure is closely connected to the surface text of the document. Together, the annotated illocutionary and rhetorical goal structures constitute the *justification structure* of a document.

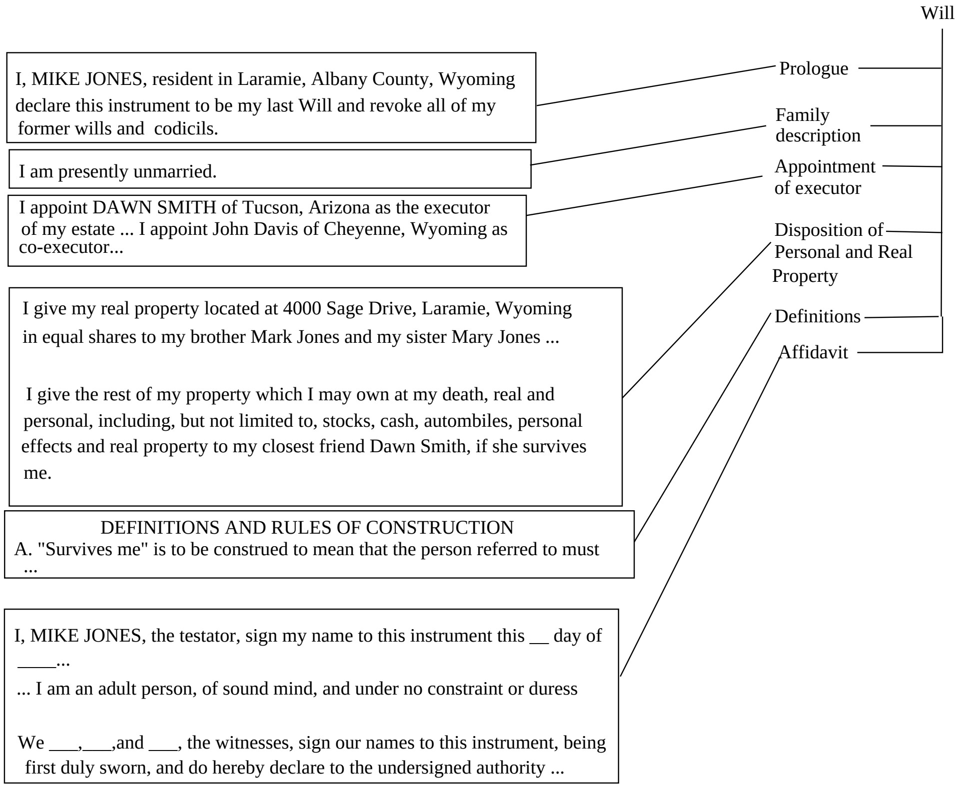

Fig. 2. The rhetorical structure of the will.

4 An Architecture for Self-Explaining Document Management

The previous sections described the tasks of document analysis, reuse, and maintenance, and argued that self-explaining documents, that is, documents containing explicit justification structures, are important for these tasks. This section proposes a *justification-based* framework for self-explaining document management. It uses the justification structures described in Section 3 as the basis for all transactions—both between the user and the system, and between all components within the system.

All communication between users and the document system transpires within the *document studio*, a workspace *cum* dialogue system in which specifications are constructed, new documents are edited, and retrieved archival documents are displayed. For each document currently under construction, the document studio maintains a *document object*, which consists of the evolving justification structure, the surface text, and the mapping between them. Before the document is complete, the surface text may be only partially reified.

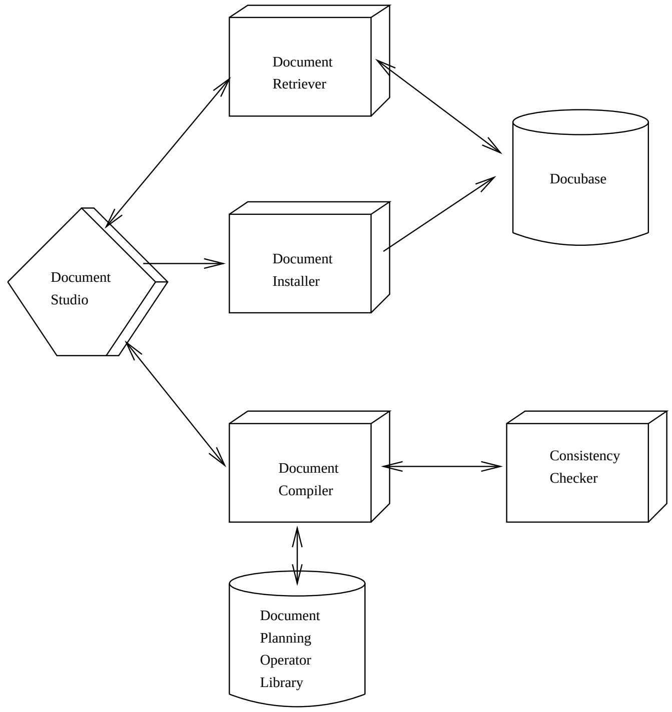

Fig. 3. An architecture for self-explaining document management.

When the user issues a request to create a new document, he or she is prompted for the document genre (*e.g.*, contracts) and sub-genre(s) (*e.g.*, employment contracts). Users may then pursue one of two construction modes: *ab initio* or adaptive. If they opt for *ab initio*, the system will engage them in a *specifications-gathering* dialogue. First, it will use the genre and sub-genre types to identify the operators in the *document planning operator library* that are relevant to the construction task at hand. Next, it will employ a hierarchical planning technique to decompose the root illocutionary goals and root rhetorical goals by querying the user. As users respond to questions about the illocutionary goals to be achieved, subsumptive operators are identified and their predicates are instantiated with user-specified values. This process bottoms out for leaf goals, which are entity and relationship variables that are instantiated with the specifics of the current document specification.

When users select the adaptive construction mode, the system engages them in a *similarity-identification* dialogue. In an iterative refinement process, it will request users to characterize the illocutionary and/or rhetorical goals to be achieved. The document studio will pass these to the *document retriever*, which will use them to locate similar documents in the *docubase*, the library of archival documents. The docubase employs justification indices that map illocutionary and rhetorical goals to document objects that achieve these goals. As a consequence, the document retriever can quickly identify archival documents whose intent is similar to that desired by a user. These documents are then presented in the document studio, and the user either deems them acceptable or annotates individual segments as being desirable or undesirable. This feedback, along with refined specifications at the goal level, is used to obtain archival documents that more closely meet the user's needs.

Once the most relevant archival documents have been located, the user adapts them to his or her current needs. Adaptation is performed by comparing the justification structure of the retrieved text to the current illocutionary and rhetorical goals. Text segments whose only purpose is to satisfy goals absent from the current situation are *excised, i.e.,* identified and removed. If the user has goals that are not satisfied by the retrieved text, retrieval is performed on other documents to find text segments that can satisfy that goal. Retrieved text segments are abstracted and reinstantiated with values appropriate for the user's goals.

Adaptive construction offers two benefits over *ab initio* construction. First, it can provide significant efficiency gains, particularly if intentionally similar documents can be located. Second, adaptation conserves justification structures across documents. This property is critical for document drafting tasks in which (1) complex large-scale (perhaps multi-volume) documents must achieve a global consistency, (2) multi-generation documents must retain a semantic (as opposed to merely a syntactic) similarity across generations, or (3) "hyper-formal" documents must make declarations that achieve complicated illocutionary goals while simultaneously adhering to strict legal or regulatory rules and attending to baroque stylistic conventions.

To permit maximum expressiveness, at any time users may compose and insert their own text segments. Though critical, this capability is problematic for inference unless the system constructs a justification (sub-)structure for the user's text. It must therefore either infer the justification structure of the newly entered text or engage in a dialogue with the user in which the illocutionary and rhetorical structure are interactively ascertained. Given the limitations of NLP, the latter alternative is adopted. Accordingly, the system integrates the new text into the justification structure by engaging in a *text characterization* dialogue in which the user is asked to categorize the text with the appropriate illocutionary and rhetorical representational vocabulary.

When users wish to see the effects of their specifications, they request a document compilation. Upon request, the document studio passes the justification structure developed up to this point—note that depending on the stage of completion, it may or may not be completely instantiated—to the *document compiler*, which employs a top-down planning mechanism to construct as much of the justification structure as the current information will permit. This structure is then passed to the *consistency checker*, which inspects the structure for constraint violations with respect to the selected genre and sub-genres. The resulting surface text, together with formatting specifications and the constraint violations, are then passed to the document studio, which displays the properly formatted document and reports any constraint violations.

At any time, users can highlight specific regions of a document and request that rationale behind the segment be explained. The document studio will inspect the justification structure associated with the document object to identify the sub-trees of the illocutionary and rhetorical structures that are associated with the specified segment. It will then display the justifications, both in tree form and in textualized form. Users may also highlight a region of the document and request to view similar text segments of archival documents. The document studio will extract the selected sub-trees of the justification structure and pass these to the document retriever, which will identify analogous segments in the docubase. Finally, users may view a document's entire justification structure. They can specify for it to be displayed in either an *abstracted* mode, where the variables remain uninstantiated, in an *instantiated* mode, where all bindings are displayed, or in a *juxtaposed* mode, where the justification structure and its mapping to the surface text are displayed side by side.

5 Example: Use of Self-Explaining Documents

Suppose that an attorney, Clarice Darrow, a general practitioner, wishes to draft a will for a client, Mary Baker, who desires to devise her home to husband, John Baker, and to provide for the guardianship for her minor children, Sally and Bill, if her husband should predecease them. Suppose that Clarice wishes to find an existing will to adapt to meet her current goals. The attorney's interaction with the system might consist of the following steps:

1. Retrieval: finding an existing document that can be adapted for the current situation.
   (a) Illocutionary goal identification. Darrow first identifies the illocutionary goals that she wishes the document to achieve. The system's interface may include a check-sheet, interviewing procedure, or domain-specific expert system to elicit goals from the user in a process of iterative refinement. The system should be able to infer any necessary subgoals of the specified goals. A simplified illocutionary goal tree is shown in Figure 4. This goal tree shows, *e.g.*, that in order to make a bequest, it is necessary for the testator to establish competency, and this in turn requires establishing three subgoals: age; mental competency; and absence of duress.
   (b) Indexing. The most closely matching existing documents under a user-modifiable similarity metric are then located and retrieved. Suppose that the will shown in Figure 1 is one of the documents retrieved.
2. Comparison. The user, Clarice, must now determine how the retrieved document should be modified to apply to the current situation. This requires comparing the illocutionary goal tree for the current situation, shown in Figure 4, with the goal tree for the retrieved document. This comparison is shown in Figure 5. This comparison indicates that the retrieved document

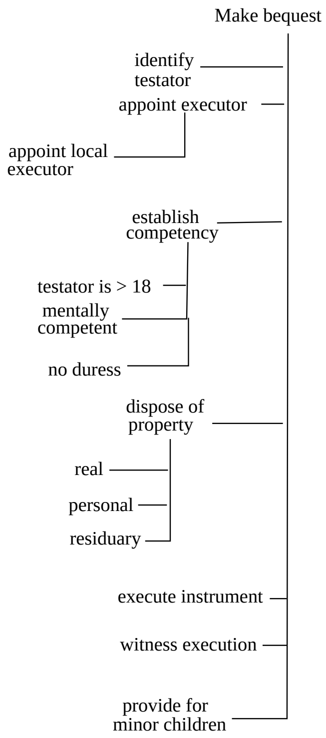

Fig. 4. An illocutionary goal hierarchy.

differs in that (1) it achieves the appointment of an executor by appointing an out-of-state executor together with a local co-executor, whereas the current goal tree satisfies this goal by appointing a local executor, and (2) the retrieved document doesn't provide for minor children.

3. Adaptation: modifying the retrieved document to satisfy the current illocutionary goals.
   (a) Excision. Text whose only purpose is to satisfy goals that aren't present in the current situation should be identified and removed. For example, text whose purpose is to appoint a local co-executor is not necessary in the current situation and is removed.
   (b) Augmentation. For each goal that does not appear in the retrieved text (such as `provide for minor children`), present the user with examples of text from other documents that can satisfy that goal. This requires retrieval not at the level of granularity of the entire document, but at a much finer granularity. In this case, just those portions of several previous wills that provide for guardianship are retrieved. For example one retrieved segment might be the following portion from the will of Eleanor Silver: "In the event that my wife, Eleanor Silver, should predecease me

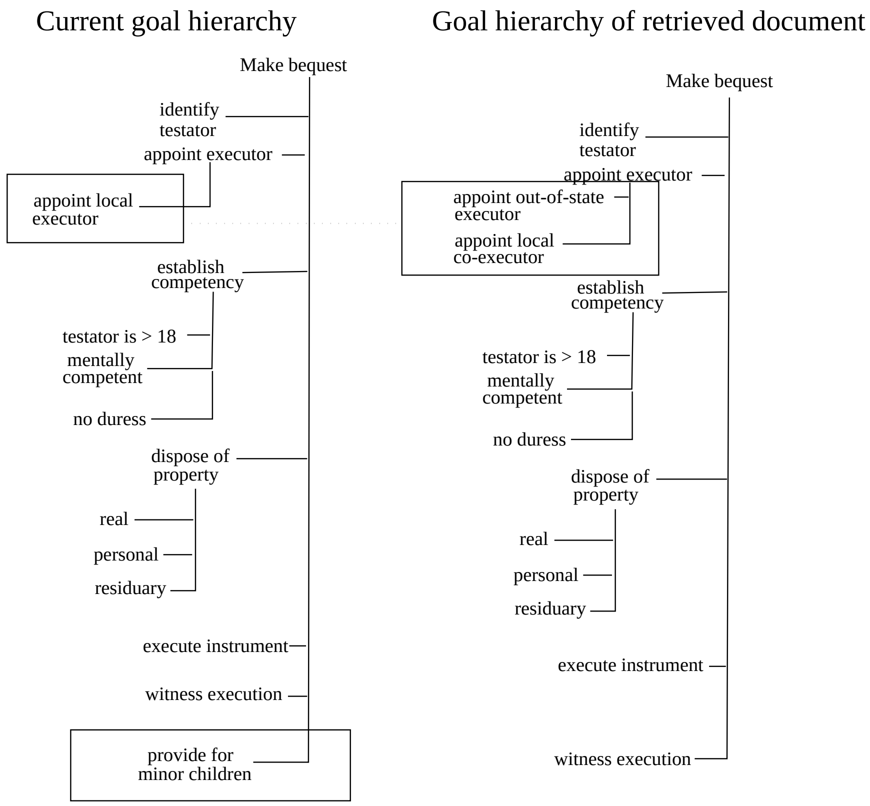

Fig. 5. A comparison of the given goals with the illocutionary structure of the retrieved document.

before my children, Sarah and Martha achieve their majority, I appoint my brother, Elmer Goldsmith, to be guardian ..." The user can view these test segments and select the most appropriate segment in view of (1) the illocutionary structure of the text segment itself, (2) any annotations attached to the text segment at the time of their creation, or (3) stylistic preferences of the user.

(c) Substitution. Since substitutable elements of all items in the docubase are tagged with type labels using standard template techniques, retrieved segments can be abstracted and reinstantiated. In the current example situation <testator: Mary Baker> to be substituted for corresponding values in the retrieved text, *e.g.*, <testator: Eleanor Silver>. The result of the substitution is a self-explaining document that satisfied the current set of illocutionary goals.

4. Storage. The final document, which may embody components from multiple previous documents, is added to the document library indexed by illocutionary and rhetorical goal structure.

6 Implementation Plans

We are currently exploring the application of justification structures to document reuse in the context of Colorado Court of Appeals show-cause orders. Jurisdictional show-cause orders are issued during jurisdictional screening, a process of determining whether the requirements for an appeal have been satisfied. If there appears to be a jurisdictional defect, a 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. Show-cause orders are produced in relatively high volume (several hundred per year), are complex enough to require drafting by an attorney, but have sufficient stylistic and substantive consistency to facilitate reuse.

Our research agenda is as follows. We are currently engaged in developing the domain theory for show-cause orders (the objects and predicates that define the illocutionary and rhetorical structures). When this process is complete, we will undertake the construction of each component of the architecture. To supply a uniform representation for all of the knowledge structures, all illocutionary structures, rhetorical structures, and planning operators will be implemented in a unification-based constraint system [Elh91]. As a result of the single representation, all adaptation procedures (document compilation and consistency checking) will be performed with a single mechanism: constraint propagation. Although this brings with it an initial development overhead, we expect to witness significant software engineering benefits in the form of rapid extensions to the domain theory. Finally, once the system for show-cause orders is complete, we will run extensive empirical evaluations to measure the gains in efficiency and correctness.

7 Related Work

Indexing and adaptation are central to any case-based approach to problem solving. Early case-based reasoning systems, *e.g.,* [Kol84, SW86, PBH90], typically addressed tasks, such as classification, that require little or no adaptation. By contrast, more recent CBR research has often addressed tasks such as design, planning, or configuration, that require complex adaptation, *e.g.,* [Vel92, BS95, SK95, BA95]. Document drafting typifies a design task requiring complex adaptation for case reuse. The research described in this paper is intended to formalize the knowledge required for document reuse, just as the projects cited above identified knowledge required for reuse of other types of design, planning, and configuration cases.

Our model of document justification structures 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.*, [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, that is, the goals that a speaker intends to accomplish through that discourse [Gri75, Sea69]. Illocutionary analysis is essential for self-explaining documents because few documents have an exclusively communicative purpose. Instead, as emphasized above, documents are often intended to elicit information, persuade, memorialize events, or to accomplish performative goals.

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 legal briefs and judicial decisions [Bra93].

The explanation community has extensively studied the process of planning and realizing text given a set of discourse specifications. Over the past decade, their work research on discourse planning [McK85, Par88, Hov90, Hov93, Caw92, Sut93, Moo95, LP96] 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 [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 automated report generation community has addressed another form of text production from an underlying domain structure: the derivation of technical documentation from program traces generated during software development or use [MRK95, KMR93]

8 Summary

Document drafting typifies a design task requiring complex adaptation for case reuse. This paper has proposed a self-explaining document framework that uses explicit knowledge of the document's illocutionary and rhetorical structure for indexing and adapation. We are currently instantiating the framework in the context of judicial show-cause orders. It is our hypothesis that, by equipping documents with a justification structure that includes both illocutionary and rhetorical components, self-explaining document systems can assist document designers with constructing, maintaining, and interpreting complex documents.

References

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

[BA95] K. Branting and D. Aha. Stratified case-based reasoning: Reusing hierarchical problem solving episodes. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence(IJCAI-95)*, Montreal, Canada, August 20–25 1995.

[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.

[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.

[BS95] B. Bartsch-Sporl. Towards the integration of case-based, schema-based and model-based reasoning for supporting complex design tasks. In *Lecture Notes in Artificial Intelligence*, pages 145–156, Sesimbra, Portugal, October 1995. Springer.

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

[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, NY, 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.

[Hov90] E. H. Hovy. Pragmatics and natural language generation. *Artificial Intelligence*, 43:153–197, 1990.

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

[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.

[Kol84] J. Kolodner. *Retrieval and Organizational Strategies in Conceptual Memory: a Computer Model*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1984.

[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.

[LP96] J. C. Lester and B. W. Porter. Scaling up explanation generation: Large-scale knowledge bases and empirical studies. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, Portland, Oregon, to appear 1996.

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

[PBH90] B. W. Porter, E. R. Bareiss, and R. C. Holte. Concept learning and heuristic classification in weak-theory domains. *Artificial Intelligence*, 45(1–2), 1990.

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

[SK95] B. Smyth and M. Keane. Experiments on adaptation -guided retrieval in case-based design. In *Lecture Notes in Artificial Intelligence*, pages 313–324, Sesimbra, Portugal, October 1995. Springer.

[Sut93] D. D. Suthers. *An Analysis of Explanation and Its Implications for the Design of Explanation Planners*. PhD thesis, University of Massachusetts, February 1993.

[SW86] C. Stanfill and D. Waltz. Toward memory-based reasoning. *Communications of the ACM*, 29(12), 1986.

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

[Vel92] M. Veloso. *Learning by Analogical Reasoning in General Problem Solving*. PhD thesis, Carnegie Mellon University, 1992.

[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.