A Generative Model of Narrative Cases

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

Abstract

Effective case-based reasoning in complex domains requires a representation that strikes a balance between expressiveness and tractability. For cases in temporal domains, formalization of event transitions in a *narrative grammar* can simplify both the user's task of problem formulation and the system's indexing, matching, and adaptation tasks without compromising expressiveness. This paper sets forth a model of temporal cases based on narrative grammars, demonstrates its applicability to several different domains, distinguishes two different similarity metrics—sequence overlap and tree overlap—and shows how the choice between these metrics depends on whether nonterminals in the narrative grammar correspond to abstract domain states or merely represent constraints on event transitions. The paper shows basic-level and legal-event narrative grammars can be used together to model how human lawyers interleave fact elicitation and analysis.

1 Introduction

A fundamental issue in the design of AI systems is the tradeoff between representational adequacy and tractability. Use of the most expressive representations, such as first-order predicate calculus, entails high costs both in computational complexity and in comprehensibility, that is, the ability of users to formulate problems and understand solutions. Conversely, the most tractable representations, such as vectors of numeric or symbol features, are severely limited in expressiveness. The most appropriate representation for a given problem is one that optimizes this tradeoff, maximizing the ability to represent the factors relevant to problem solving in the domain without sacrificing efficiency or comprehensibility. Finding this optimal tradeoff requires exploiting the regularities in the structure of the domain.

This paper proposes a formalism, termed a *narrative grammar*, for exploiting the temporal structure of narrative cases, that is, cases that involve sequences of events. The paper shows how narrative grammars can improve both computational complexity and comprehensibility while maintaining a high level of representational adequacy. The paper first sets forth the central characteristics of narrative cases. Section 3 then describes how the constraints on event transitions in narrative cases can be expressed by narrative grammars, and Section 4 describes how narrative grammars can assist problem formulation and similarity assessment. The range of applicability of the narrative grammar approach is then discussed in Section 5.

2 Narrative Cases

Case-based reasoning is a very general computational mechanism. The range of possible cases is as wide as the ranges of reusable solutions and examples. However, in many domains, such as legal reasoning and process control, cases typically involve sequences of events. For example, the rights and liabilities of persons under contract law depend not just on the statements and actions performed by the persons, but also on the order and timing of the statements and actions. The order and timing of events is also essential for many other areas of law, such as negotiable instruments and workers compensation law.

Two key criteria for representational adequacy of narrative cases can be distinguished: *horizontal extensibility* and *vertical extensibility*. These terms are suggested by the venerable representation conventions of indicating abstraction by position along a vertical axis and indicating time by position along a horizontal axis. *Horizontal extensibility* is the ability to express sequences of actions of arbitrary length. A case representation consisting of a template (or, equivalently, a frame, feature vector, or script) to be instantiated by the facts of a particular case is incapable of representing longer sequences of events than those anticipated in the design of the template. In view of the unpredictable variety of possible narratives that can give rise to cases, it is often infeasible to devise a template capable of accommodating all possible narratives.

The second criterion of representational adequacy, *vertical extensibility*, is the ability to represent case facts in sufficient detail to express all relevant aspects of the case. For example, a representation that can express only that a person "agreed" or "refused" an offer is incapable of even expressing a problem in which the issue is whether a statement was a sufficiently definite "manifestation of assent" to constitute a valid acceptance. Vertical extensibility is the absence of arbitrary restrictions on the degree of expressible details.

First-order predicate calculus (FOPC), semantic networks, and other general relational languages, such as KL-1 [BS85] and Classic [BMPS+91], are characterized by both horizontal and vertical extensibility. This is because these languages impose no constraints on either the length of event sequences or the granularity with which events are described. However, general relational languages give rise to three distinct costs.

The first, and probably most important, cost associated with highly expressive representations is the cost of *problem formulation*, the expression of a problem in a form amenable to manipulation by a computer. In systems that use featural representations of cases, problem formulation is typically quite straightforward, requiring only assignment of values to each case slot. Problem formulation in a relational representation, by contrast, may be complex, time-consuming, and prone to error and inconsistency. For example, representing even a simple story in first-order predicate calculus is a challenging undertaking even for an experienced logician.[1]

The second cost is in difficulty of retrieval. Retrieval is relatively straightforward when cases are represented as feature vectors because individual case features are meaningful indices to such cases. By contrast, no general retrieval mechanism for cases represented using a relational language has been identified that is guaranteed both to be accurate and significantly faster than exhaustive matching [Bra92].[2]

Third is the cost of case matching. The cost of determining the degree of match between two cases represented as feature vectors is proportional to the number of features in each case. If cases are represented using a relational formalism, by contrast, determining the degree of match requires determining the size of the largest common subgraph. This problem is, in general, exponential in the size of the graphs [GJ79].

In summary, narrative cases require a representation with both horizontal and vertical extensibility. Relational representations satisfy this requirement, but impose high costs in matching, retrieval, and problem formulation.

3 Narrative Grammars

In this paper, a set of relevant states in a given domain, together with a set of productions expressing states transitions that are meaningful in the domain, is termed a *narrative grammar*. In a narrative grammar intended to be used for case elicitation, the states should correspond to "basic-level" [RMG+76, Lak87] facts, *i.e.*, facts at the level of abstraction that occurs most frequently in ordinary discourse.

In the domain of contract formation, for example, relevant events can consist of statements (*e.g.,* proposals, refusals, agreements, and retractions) or transfers (*e.g.,* unilateral or reciprocal payments, deliveries, or performance of services). Other types of events are, in general, irrelevant. These basic level events are distinct from *legal events*, such as offers and acceptances. For example, a recurring issue in contract formation is whether a purported agreement with terms that differ from the original proposal is an acceptance or a counteroffer. This issue is only expressible if there is a distinction between the basic level at which proposals, agreements, and other actions are expressed, and a more abstract level of legal predicates, such offers and acceptances. Similarly, a narrative grammar for negotiable instruments would include states corresponding to actions such as drafting, endorsing, rejecting, losing, stealing, or paying checks or other negotiable instruments.

A grammatical formalism is well-suited for narrative cases because narrative case have a recursive constituent structure For example, one can reach a state in which an offer is pending either from a different state through an offer, or through an offer-pending state through a counter-offer. There are no *a priori* limits on the number of offers and counter-offers that can precede an acceptance. This recursive, potentially unbounded structure is easily modeled with a grammar, but is modeled only with difficulty through alternative formalisms, such as scripts or semantic networks.

The benefit of using a narrative grammar for case fact elicitation is that the generative character of a grammar permits horizontal extensibility: a case can consist of an arbitrarily long sequence of events.[3] The user is permitted to tell his or her own story, without having to fit it into a preexisting template. Moreover, as shown in the next section, the computational expense of retrieval and matching case facts parsed by a narrative grammar is minimal in comparison to the costs associated with relational representations.

3.1 Antecedents of Narrative Grammars

The narrative-grammar model is based on two previous ideas. The first is *story grammars* [AS90] a formalism originally devised by anthropologists and linguists to account for regularities in folk tales and myths. Story grammars express ordering constraints on the events that occur in folklore and are intended to capture psychological or linguistic regularities. By contrast, the purpose of narrative grammars is to assist case formulation, indexing, and comparison by (1) organizing the elicitation and structure of case facts around a temporal ordering and (2) restricting the facts elicited to only those meaningful in the domain while (3) permitting horizontal extensibility.

A second antecedent idea is Anne Gardner's augmented transition network (ATN) for contract formation [Gar87] pp. 121–122. This ATN makes explicit the set of legally relevant states that can be occupied during the process of contract formation, e.g., "offer pending","offer accepted", and the legal events that lead to a transition from one state to another, e.g., an acceptance in the state "offer pending" leads to a transition to "offer accepted". However, as conceived in this paper, narrative grammars are not restricted to transitions among legally relevant states. Indeed, if a narrative grammar is used in the elicitation of case facts, the grammar must apply to transitions among basic-level facts, and not merely legally relevant states. Thus, Gardner's contract formation ATN is a special case of the general category of narrative grammars.

The productions in a narrative grammar are limited to events that are meaningful in the grammar's domain. For example, in a basic-level narrative grammar for contract formation, any party can make a proposal at any time: a seller might follow one solicitation with another, or a buyer might answer a proposal with a counterproposal. However, it would not be meaningful for a person to withdraw a proposal if no proposal has been made.

3.2 Implementation of Narrative Grammars

One possible representation for narrative grammars is as Definite Clause Grammars (DCG). The examples that follow are from NP1 (Narrative Parser 1), a prototype system

[1] See [Dav90] for an extensive exploration of these complexities.

[2] However, retrieval techniques satisfying these criteria have been devised for specialized types of graphs [Coo89]. An anytime graph retrieval algorithm is described in [Sch95], and numerous efficient approximate algorithms have been developed, e.g., [MB98].

[3] Narrative grammars do not, *per se*, address the problem of vertical extensibility.

developed (in Prolog) by the author for case elicitation using narrative grammars.

Event Representation. Individual events must be represented in sufficient detail to distinguish actors, recipients, instrumentality, and other roles. Objects, tailored on Fillmore's case frames [Fil68], provide a natural representation at the granularity of basic level facts while making the needed role information explicit. For example, a first approximation of a proposal event frame might be as follows:

```
proposal(proposer(P1),
         proposee(P2),
         thing_proposed(
           exchange(
             transfer1(T1),
             transfer2(T2))),
         instr(I),
         proposal_time(T)
```

State Representation. The state resulting from a proposal action could be represented in a similar way. For example, the state that results from the execution of a `propose` event would include the following:

```
proposal_pending(proposer(P1),
                 proposee(P2),
                 thing_proposed(
                   exchange(
                     transfer1(T1),
                     transfer2(T2))),
                 proposal_initiation_time(T))),
```

The representation of a state in general requires several frames. In the contract formation domain, for example several proposals may be pending simultaneously.

State Transitions. State transitions are expressed as <State-frame,Event,State-frame> triples. For example, the following triple represents the transition from a state in which a proposal is pending to a state in which an agreement apparently exists, triggered by an acceptance.

```
transition(proposal_pending(
             proposer(P1),
             proposee(P2),
             proposed_exchange(
               exchange(
                 transfer1(T1),
                 transfer2(T2))),
             proposed_time(T)),
           acceptance(
             acceptor(P1),
             acceptee(P2),
             instr(I),
             proposal_agreed_to(
               exchange(
                 transfer1(T1),
                 transfer2(T2)))),
           agreement(party(P1),
                     party(P2),
                     action(exchange(
                             transfer1(T1),
                             transfer2(T2))),
                     time(T)))
```

Event elicitation. In NP1, the user is presented at each stage of the elicitation of the case facts with a set of events corresponding to meaningful transitions from the current state. Variable bindings specified by the transition are presented as defaults, but the user has the option of providing different variable bindings. Here, for example, the query that is posed in a state in which John has offered to sell Mary one hundred tons of coal for one thousand dollars. NP1 lists the events that are meaningful at this stage, together with the variable bindings specified by the transitions:

```
Please select the next action

1. rejection(agent(mary),
          proposal_rejected(
          exchange(
             transfer1(one_hundred_tons_coal),
             transfer2(one_thousand_dollars))))
2. acceptance(acceptor(john),
           acceptee(mary),
           proposal_agreed_to(
             exchange(
               transfer1(one_hundred_tons_coal),
               transfer2(one_thousand_dollars))))
3. proposal(proposer(mary),
         proposee(john),
         proposed_exchange(
           exchange(transfer1(_G1624),
                    transfer2(_G1626))),
         proposed_time(_G1627))
4. expiration(proposed_exchange(
                exchange(
                  transfer1(one_hundred_tons_coal),
                  transfer2(one_thousand_dollars))),
           expiration_time(_G1628))
5. other_action(agent(john),
             recipient(mary),
             action(_G1559),
             time(jan_1_1998))
6. end

[1..6]?
```

After selecting an event (e.g., acceptance), the user is queried on each variable in the event frame. The user can accept some variable (e.g., John, Mary), and alter others (e.g., the purchase price).

NP1 returns a list of event/state pairs representing the relevant facts of a narrative. The complexity of narratives elicited by NP1, as measured by the equivalent number of ground sentences, is comparable to the cases used by GREBE [Bra91a] or in the ACME experiments in analogical reasoning [HT89]. However, NP1 elicits the narrative in just a few minutes, as compared with the hours necessary to specify a case as an unstructured set of tuples required for GREBE or ACME. Moreover, the opportunities for inconsistencies in case-fact representation between users and system builders, different users, or even the same user on different occasions, is drastically reduced. This increased speed and consistency of case-based representation does not come at any cost horizontal extensibility, because NP1 doesn't impose any restrictions on the length of the sequence of events constituting the case nor on the relationships among variable bindings in that sequence.

A practical narrative graph fact-elicitation interface would need to conceal from the user the details of the representation of event and state structure, e.g., through a graphical user interface that presented a dialog box for each variable to be instantiated and text templates so that instantiated frames could be presented in natural language text.

```
New Case:   E1 E2 E3 E4 E5 -- --
Precedent:  E1 E2 E3 E4 E5 E6 E7
```

Figure 1: The new case is a proper subsequence of the precedent.

```
New Case:   E1 E2 E3 E4 E5 E6 E7
Precedent:  E1 E2 E3 E4 E5 -- --
```

Figure 2: The precedent is a proper subsequence of the new case .

4 Matching and Retrieving Narrative Cases

Several potential uses of narrative case facts elicited through a narrative grammar can be distinguished. In the simplest case, facts can be used simply as indices for retrieval of cases or other authorities. For example, a practical barrier to the use of conceptual retrieval systems like the negotiable instruments system described in [Haf81] is that they require to user to pose the query in an unfamiliar formalism. A negotiable instruments narrative grammar could enable users to simply describe the sequence of events giving rise to their query. If a parsed narrative were associated with each relevant precedent or other authority, these authorities could then be retrieved by comparing the user's narrative to each and presented to the user in order of similarity, without further analysis. Alternatively, the relationship between the user's narrative and the narrative associated with a case could form the basis of analysis of the case and, potentially, additional fact elicitation.

4.1 Similarity Assessment for Narrative Cases

The simplest similarity assessment metric for narrative cases is sequence matching [SM97]. However, unlike molecular biology (the field that has given rise to most sequence-matching algorithms), where the elements of the sequences being compared have no internal structure relevant to the comparison, event sequences produced by narrative grammars have internal variable bindings which must also match. For example, two sequences consisting of a `proposal` followed by an `agreement` are not identical if the terms of the `agreement` are identical to those of the `propose` in one sequence but not the other. One approach to comparison is therefore to perform it in two stages: sequence matching followed by assessment of variable binding similarity.

4.1.1 Sequence Matching

A perfect match between event sequences indicates identical cases, but the subsequence and super-sequence relations

```
New Case:   E1 E2 E3 E4 E5 E6 E7
Precedent:  E1 E2 E3 -- -- E6 E7
```

Figure 3: The best match between event sequences requires introduction of a gap.

```
New Case:   E1 E2 E3 -- -- E6 E7 E8
Precedent:  -- E2 E3 E7 E5 E6 -- E8
```

Figure 4: The best match requires introduction of gaps in both sequences.

```
New Case:
          S0   S1   S2   S3          S4   S5
            E1   E2   E3   E4   --   E6   E7

Precedent:
          S0   S1   S2   S3          S4   S5
            E1   E2   E3   --   E5   E6   E7
```

Figure 5: If events E4 and E5 are irrelevant, the state sequences match despite differing occurrences of these events.

between a new case and a precedent, illustrated in Figures 1 and 2, are indicative of a high degree of match as well. Figure 3 illustrates a situation in which the new case has a subsequence of events absent from the precedent. Such differences are likely to be quite significant unless events E4 and E5 are known to be irrelevant. Figure 4 illustrates a situation where the new case and precedent share events, but the best match requires introduction of gaps into both sequences.

How can matches of these various types be quantified? Standard dynamic programming algorithms are parameterized by the penalties assigned to gaps and mismatched items. The precise penalties that should be assigned depend on the particular domain, but once these penalties have been determined, the optimal alignment between two event sequences can determined in quadratic time.[4] Tractable retrieval of the best matching sequence from a database of sequences can be accomplished by algorithms such as BLAST or FAST [SM97].

The retrieval process can accommodate narrative grammars that have been designed to include events that are irrelevant and therefore do not result in state changes. One approach is to alter the similarity metric so that there is no cost associated with a match between an irrelevant action and a gap. Alternatively, the matching can be performed between states rather than events. As illustrated in Figure 5, either of these approaches leads to a perfect match between narrative cases that differ only in irrelevant events.

4.1.2 Variable Binding Comparison

As stated above, sequences of identical events may represent very different narratives if there is a mismatch between the variable bindings in the two sequences. Ideally, a measure of the largest possible set of consistent variable bindings should be included in the similarity metric used in sequence matching. However, this would render the matching algorithm intractable. Even determining the best possible mapping between the variables in two sequences is potentially

[4] Note, however, that this would constitute a static, global similarity metric. An interesting technical question is how to incorporate context-sensitive similarity metrics into this quadratic matching procedure. The answer to this question depends, of course, on the nature of the context-sensitive similarity criterion.

intractable, requiring in the worst case consideration of every possible mapping between variable bindings.

However, in practice, variable mappings can be considered in a greedy, event-by-event fashion. For example, consider comparing two contract formation cases that start with the following events:

```
proposal(proposer(john),
         proposee(mary),
         thing_proposed(
           exchange(
             transfer1(one_hundred_tons_coal),
             transfer2(one_thousand_dollars))),
         instr(telephone),
         proposal_time(T1)

proposal(proposer(bill),
         proposee(sally),
         thing_proposed(
           exchange(
             transfer1(db_software),
             transfer2(five_thousand_dollars))),
         instr(fax),
         proposal_time(T2)
```

The only mappings that ordinarily need be considered are mappings that are extensions of the mapping needed to match the two initial events:

> john → bill
> mary →sally
> one hundred tons of coal → db software
> one thousand dollars→ five thousand dollars
> telephone→ fax
> T1 → T2

In general, the only alternative mappings that need to be considered between any pair of events are those that are consistent with the best mappings between the corresponding events in previous events. The set of alternative variable matchings that must be considered at each step is bounded by the number of variables that can occur in any individual event.

4.1.3 Tree Matching

The forgoing discussion has assumed that nonterminal nodes in a narrative grammar parse tree have no inherent meaning and that the relevant aspect of cases for comparison is therefore event sequences. However, in some cases the domain theory underlying a narrative grammar may include abstract states corresponding to nonterminals. Under these circumstances, comparison between two narrative case parses should be in terms of parse trees rather than between event sequences.

Comparison between trees is, in general, very inexpensive as long as a mismatch between nodes at one level of abstraction is guaranteed to obviate comparisons between the descendents of those nodes. Under these circumstances, comparison of two trees is linear in the size of the trees. Various criteria have been used for assessing similarity between hierarchical structures, e.g. [Bra91a, BS98]. In general, these schemes assign to the match between two nodes a weight inversely proportional to their distance from the root.

Figure 6 represents schematically two partially matching narrative parse trees. In the left subtrees (each rooted at S2) the leaf nodes must be compared, but since S3 and S4 do not match, no comparison of the leaves of the right subtrees is needed.

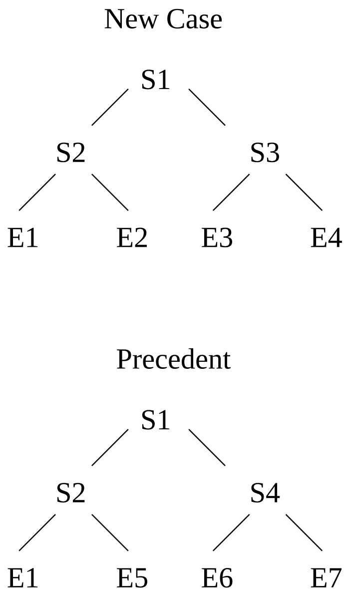

Figure 6: The two narrative parse trees match at states S1 and S2 and event E1. The subtrees rooted at S3 and S4 need not be compared.

4.2 Grammar Hierarchies and Narrative Precedents

A narrative grammar for fact elicitation should consist of events at the basic level, corresponding to ordinary discourse. However, Gardner's contract-formation ATN illustrates how the narrative structure of the legal predicates in a domain can also be expressed in a narrative grammar. The observation that both basic level and legal-event narrative grammars are possible raises the issue of the relationship between grammars of these two types.

In domains like contract formation, the transitions in a legal-event narrative grammar are a proper subset of the transitions in the basic-level grammar. This is because in a legal-event narrative grammar only legally effective events, e.g., acceptance of an offer, are capable of effecting a transition from one state to another, e.g., from pending offer to contract. At the basic level, however, any transition is possible that makes sense in the domain, regardless of whether it is legally effective. For example, a person can `agree` to a `proposal` regardless of whether the proposal constitutes a legally effective offer or whether the agreement constitutes an acceptance or a counteroffer.

Thus, for a domain with both basic-level and legal-event narrative grammars, it makes sense to elicit a set of facts via the basic-level grammar and then parse the resulting event sequence with the legal-event grammar. In general, no unique parse of the basic-level event sequence is possible. This is because the problem of open-texture means that there is uncertainty about whether a given basic-level event satisfies the legal predicate corresponding to a transition between legal states.

Gardner's system [Gar87] used heuristics to attempt to determine whether a legal predicate was clearly satisfied—and a transition between legal states clearly effected—or

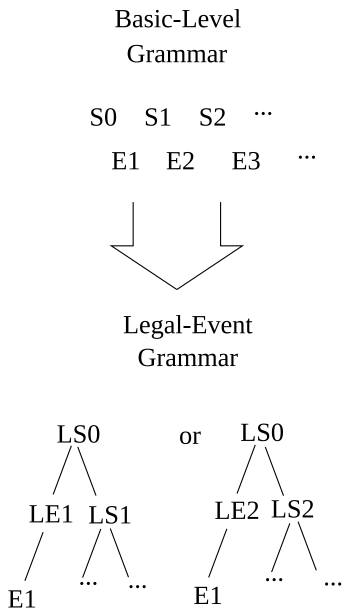

Figure 7: An event sequence elicited through a basic-level narrative grammar and then parsed with an event-level grammar.

whether there was a "hard question". In the latter case, the system simply branched. Subsequently, enumerable research projects have addressed the task of reasoning about whether an open-textured legal predicate has been satisfied, based on, e.g., case-based reasoning [Ash90, Bra91b, SR92], theory coherence [McC95], merit arguments [Pal97], etc.

When a human attorney analyzes a client's narrative, the attorney attempts to elicit additional details about the events that give rise to "hard" issues. A secondary parse of a basic-level event narrative by a legal-event grammar provides a mechanism for focusing inquiry on the key factual issues of the case and ordering this inquiry in an intuitive fashion.

Figure 7 shows schematically a situation in which a basic-level narrative grammar has been used to elicit a series of basic-level events E1 E2. . . . When this initial fact elicitation is completed, a legal-event grammar is used to parse the basic-level events into legal events (LE1 . . .) and legal states (LS0 . . .). In this case, there is uncertainty whether basic-level event E1 constitutes legal event LE1 or legal event LE2. Accordingly, there are two alternative partial parse trees.

Figure 8 shows a case library containing two cases represented as legal-event parses. In precedent P1 legal event LE1 was satisfied by basic-level event E$'$, where in precedent P2 legal event LE2 was satisfied by basic-level event E$''$. A determination whether E1 is more like LE1 or LE2 can be assisted by eliciting additional facts concerning the similarities and differences between E1 and each of E$'$ and E$''$. Standard CBR techniques can be used for this determination.

In general, elicitation of additional facts may not be sufficient to determine a unique legal-event parse if E1 presents a "hard" issue. Thus, several alternative parses may be found. However, the use of both basic-level and legal-event grammars leads to a structured interaction in which (1) the relevant events giving rise to the legal question are elicited in a narrative fashion, and (2) elicitation then focuses on

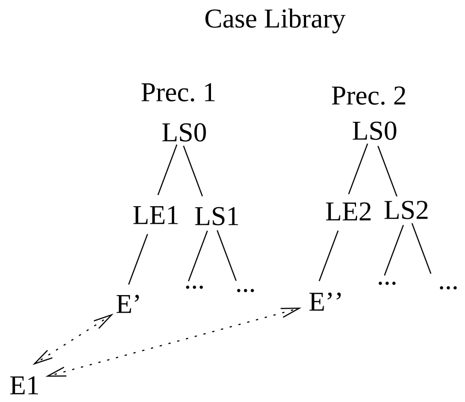

Figure 8: A case library containing two cases represented as legal-event parses. Since there is uncertainty concerning whether E1 satisfies legal event LE1 or legal event LE2, E1 is compared to examples of each, i.e., E$'$ and E$''$ respectively.

the details necessary for the best analysis possible given the tools for reasoning and arguing about open-textured concepts.

5 Related Work

As discussed in Section 3, the two direct antecedents of narrative grammars are Anne Gardner's augmented transition network for contract formation [Gar87] and *story grammars* [AS90]. A number of previous projects have addressed the problem of representing cases with temporal structure, e.g., scripts [SA77], the dynamic memory model [Sch82, Kol84], multicases [ZWA93], semantic networks [Bra91a], etc. Other work has addressed the requirements for a representation of temporal relationships adequate for legal reasoning, e.g., [McC89].

However, none of this research has addressed the focus of the narrative-grammar approach, which is use of temporal structure to simplify problem formulation, retrieval, and matching without sacrificing horizontal extensibility.

6 Conclusion

This paper has proposed the use of narrative grammars as a mechanism for achieving a balance between expressiveness and tractability in representation of cases with a temporal structure. For such cases, narrative grammars can simplify both the user's task of problem formulation and the system's indexing, matching, and adaptation tasks without compromising expressiveness.

Basic-level narrative grammars can restrict elicitation to only those facts that are potentially relevant without biasing the elicitation towards any particular legal analysis. Parsing the resulting event sequence with a legal-event grammar can then focus elicitation on details of the narratives important to analysis of the "hard" issues presented by the case, and the parses can be used to index precedents with similar legal-event sequences.

The scope of the narrative-grammar approach to case elicitation depends on a number of issues:

Scaling up. NP1 is simply a prototype designed to illustrate the feasibility of narrative grammars. Designing a complete narrative grammar for a domain such as contracts, negotiable instruments, or workers compensation law, is a task of a magnitude comparable to designing a representation in any other expressive formalism, i.e., a sizable undertaking. However, only after narrative grammars have been developed for several domains can the difficulty of scaling up document grammars be assessed.

Interface. NP1 has a primitive command-line interface. A graphical user interface with a more intuitive design, including dialog boxes for variables and natural language queries, would greatly improve the usability of the narrative grammar approach. The best approach to the design of such an interface is a nontrivial HCI question.

Background and setting. Story grammars included not just productions, but also background and setting. Although this aspect of narratives has not been discussed in this paper, in many domains significant parts of the facts relevant to a case are not narrative in character. For example, in a workers compensation case, the relevant facts include not just the story of how a worker was injured, but also details of the contract and conventions of the industry, facts that are static with respect to the narrative. The best way to elicit background information and use this information in narrative parsing and case similarity assessment is an open question.

Design policies. For basic-level narrative grammars, two conventions are possible concerning irrelevant states. A parsimonious approach to irrelevant states would be to make it impossible for the user to enter facts that could not possibly be relevant. An alternative approach would expand the grammar to permit non-state changing events.

Similarly, two conventions are possible concerning multiple states. Under one approach, an `agree` event would be possible in any state reached after a `propose` event (since, e.g., someone might purport to agree to a proposal even after an alternative agreement has already been consummated). Alternatively, transitions could be governed by sets of states, corresponding to all prior events. Thus, after a `propose` and an `agree` there would be a multi-state consisting of a pending proposal and an agreement. Thus, it would be possible for the user to select transitions from either the pending proposal (e.g., the proposee could send a refusal or a counter-proposal) or the agreement (a performance of an obligation under the agreement).

Prefix matching vs. event matching. In the example of case indexing set forth above, indexing was by a single legal event. An alternative approach would be to index precedents by legal event prefixes, i.e., the match between the sequence of legal events parsed so far and the prefixes of existing cases. Indexing in this fashion would permit narrative context to be incorporated into the comparison. The precedents retrieved at each stage of legal-event parsing would be those that unfolded precisely the same way as the current case.

In domains in which narrative context is important for similarity assessment, this approach would represent an improvement of matching on the basis of single legal events. However, in some cases the details of the event prefixes are clearly irrelevant. For example, the exact number of offers and counteroffers that have occurred in a contract formation precedent may be irrelevant to the use of the last counteroffer in the series as an example of a "significant change in terms". Under these circumstances, comparisons of prefixes might actually interfer with matching to the most relevant case.

Narrative grammars illustrate how a regularity underlying cases of a particular type can be exploited to reduce the complexity of problem formulation and case manipulation with sacrificing expressiveness. If highly expressive case representations are to become the norm among legal reasoning systems, other such regularities must be identified and exploited.

Acknowledgments

This research was supported in part by a German-American Fulbright Commission Senior Scholar grant, a University of Wyoming Flittie sabbatical award, and the University of Kaiserslautern Center for Learning Systems and Applications.

References

[AS90] Sandra Andersen and Brian M. Slator. Requiem for a theory: the story grammar story. *Journal of Experimental and Theoretical Artificial Intelligence (JETAI)*, 2(3):253–275, 1990.

[Ash90] K. Ashley. *Modelling Legal Argument: Reasoning with Cases and Hypotheticals.* MIT Press, Cambridge, Massachusetts, 1990.

[BMPS+91] R. Brachman, D. McGuinness, P. Patel-Schneider, L. Resnick, and A. Borgida. Living with classic: When and how to use a KL-ONE-like language. In John Sowa, editor, *Principles of Semantic Networks: Explorations in the representation of knowledge*, pages 401–456. Morgan-Kaufmann: San Mateo, California, 1991.

[Bra91a] K. Branting. Building explanations from rules and structured cases. *International Journal of Man-Machine Studies*, 34:797–837, 1991.

[Bra91b] K. Branting. *Integrating Rules and Precedents for Classification and Explanation: Automating Legal Analysis.* PhD thesis, University of Texas at Austin, 1991.

[Bra92] K. Branting. A model of the role of expertise in analog retrieval. In *Proceedings of the Fourteenth Annual Conference of the Cognitive Science Society*, Bloomington, Indiana, July 29–August 1 1992.

[BS85] R. J. Brachman and J. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2), 1985.

[BS98] R. Bergmann and A. Stahl. Similarity measures for object-oriented case representations. In B. Smyth and P. Cunningham, editors, *Lecture Notes in Artificial Intelligence*, volume 1488, pages 25–36, 1998.

[Coo89] D. Cook. ANAGRAM: An analogical planning system. Technical report, Department of Computer Science, University of Illinois, 1989.

[Dav90] E. Davis. *Representations of Commonsense Knowledge.* Morgan Kaufmann Publishers, Inc., Palo Alto, 1990.

[Fil68] C. Fillmore. The case for case. In *Universals in Linguistic Theory*, pages 1–90. Holt, Rinehart & Winston, New York, 1968.

[Gar87] A. Gardner. *An Artificial Intelligence Approach to Legal Reasoning*. Bradford Books/MIT Press, Cambridge, MA, 1987.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability*. Freman, New York, 1979.

[Haf81] C. Hafner. *An Information Retrieval System Based on a Computer Model of Legal Knowledge*. UMI Research Press, 1981.

[HT89] K. Holyoak and P. Thagard. Analogical mapping by constraint satisfaction. *Cognitive Science*, 13(3), 1989.

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

[Lak87] G. Lakoff. *Women, Fire, and Dangerous Things*. University of Chicago Press, Chicago and London, 1987.

[MB98] B.T. Messmer and H. Bunke. A new algorithm for error tolerant subgraph isomorphism detection. *PAMI*, 20(5):493–504, May 1998.

[McC89] L. T. McCarty. A language for legal discourse 1. Basic features. In *Proceedings of the Second International Conference on Artificial Intelligence and Law*, Vancouver, B.C., June 13-16 1989.

[McC95] T. McCarty. An implementation of Eisner v. Macomber. In *Fifth International Conference on Artificial Intelligence and Law*, pages 276–286, College Park, MD, May 21–24, 1995. ACM Press.

[Pal97] J. Palmer. Legal merit arguments, legal semiotics, and the design of legal knowledge-based systems. In *Proceedings of the Sixth International Conference on Artificial Intelligence and Law (ICAIL-97)*, pages 198–206, University of Melbourne, Melbourne, Australia, June 30–July 3 1997. ACM Press.

[RMG+76] E.H. Rosch, C.B. Mervis, W.D. Gray, D.M. Johnson, and P. Boyes-Braem. Basic objects in natural categories. *Cognitive Psychology*, 8:382–439, 1976.

[SA77] R. Schank and R. Abelson. *Scripts, Plans, Goals, and Understanding*. Lawrence Erlbaum Associates, 1977.

[Sch82] R. C. Schank. *Dynamic Memory, A Theory of Reminding and Learning in Computers and People*. Cambridge University Press, Cambridge, England, 1982.

[Sch95] J. Schaaf. Fish and sink: An anytime-algorithm to retrieve adequate cases. In *Lecture Notes in Artificial Intelligence*, pages 538–547, Sesimbra, Portugal, October 1995. Springer.

[SM97] J. Setubal and J. Meidanis. *Introduction to Computational Molecular Biology*. PWS Publishing Co., 1997.

[SR92] D. Skalak and E. Rissland. Arguments and cases: An inevitable intertwining. *Law and Artificial Intelligence*, 1(1), 1992.

[ZWA93] R. Zito-Wolf and R. Alterman. A framework and an analysis of current proposals for the case-based organization and representation of procedural knowledge. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, 1993.