The Fourth International Conference on

# Artificial Intelligence and Law

## Proceedings of the Conference

**June 15-18, 1993**
**Computer/Law Institute**
**Vrije Universiteit Amsterdam**
**The Netherlands**

**Sponsored by**

***The International Association for Artificial Intelligence and Law***

***Faculty of Law, Vrije Universiteit Amsterdam***

**In co-operation with**

***ACM SIGART***

**Additional Sponsors:**

***IBM Nederland NV, Hewlett Packard, Kluwer Academic Publishers, Kluwer Law and Taxation Publishers, Kluwer Datalex, The City of Amsterdam, The Royal Dutch Academy for Sciences***

# BankXX : A Program to Generate Argument through Case-Base Search

Edwina L. Rissland, David B. Skalak and M. Timur Friedman

Department of Computer Science
University of Massachusetts
Amherst, MA 01003

{rissland, skalak, friedman}@cs.umass.edu

**Abstract**

*In this paper we describe a system, called BankXX, which generates arguments by performing a heuristic best-first search of a highly interconnected network of legal knowledge. The legal knowledge includes cases represented from a variety of points of view—cases as collections of facts, cases as dimensionally-analyzed fact situations, cases as bundles of citations, and cases as prototypical factual scripts—as well as legal theories represented in terms of domain factors. BankXX performs its search for useful information using one of three evaluation functions encoded at different levels of abstraction: the domain level, an "argument-piece" level, and the overall argument level. Evaluation at the domain level uses easily accessible information about the nodes, such as their type; evaluation at the argument-piece level uses information about generally useful components of case-based argument, such as best cases and supporting legal theories; evaluation at the overall-argument level uses factors, called argument dimensions, which address the overall substance and quality of an argument, such as the centrality of its supporting cases or the success record of its best theory. BankXX is instantiated in the area of personal bankruptcy governed by Chapter 13 of the U.S. Bankruptcy Code, which permits a debtor to be discharged from debts through completion of a court-approved payment plan. In particular, our system addresses the requirement that such Chapter 13 plans be "proposed in good faith."*

## 1. Introduction

BankXX is an experiment in bottom-up argument creation. The program searches a knowledge base for information that could provide the building blocks for an argument, and in gathering that information creates an argument from the ground up. In the absence of a top-down specification of a desirable argument, a legal researcher who is not familiar with a domain (or who is just stumped) might rely on this approach: see what can be dug up and try to piece together an argument from what is found. The intuitive motivation for this approach comes from our personal experience in legal research.

The perspective we take in BankXX complements previous work in which we sought to recognize and apply structures of legal argument imposed from the top down [Skalak & Rissland, 1992]. In a complete picture, we believe that argument generation includes a flexible control strategy, combining top-down, bottom-up and island-driving [Erman *et al.*, 1980] strategies (see also [Stucky, 1986]). In this paper, we consider a distinctly bottom-up approach.

Specifically, there are two facets to our informal view of how legal research is done in an unfamiliar area. The first is a very general sense of the kinds of knowledge that one has in an area where one is not an expert. The second facet, related to the first, is how one can use search to exploit those types of knowledge.

**Three Kinds of General Knowledge.** Even though one may not know what form an argument should take, one often has general notions, gathered from experience in doing legal research and making arguments, about:

(a) what types of domain knowledge exist (e.g., cases, legal theories), and how they are interconnected in library and resource materials (e.g., one case cites many others, a case applies a legal theory);

(b) what basic pieces are needed to make an argument (e.g., a legal theory, supporting cases) and a sense of how they fit together in an argument, particularly how they reference each other and support inferences (e.g., supporting cases may give rise to justifying analogies, contrary cases can be distinguished); and

(c) what makes an argument a good one. For instance, to the extent that an argument uses central cases it is better than one using outlying cases; to the extent that the supporting cases fit under one theory, an argument is better than one where a variety of theories must be cobbled together, *etc.*

When one does have previous experience in the specific legal area, additional knowledge can be used to constrain the research, such as knowledge of the structure of previous winning and losing arguments or special knowledge of the area, such as standards of community practice.

**Search of Available Domain Knowledge.** The approach to information gathering in BankXX is similar to what a junior associate in a large law firm might do when charged with the task of providing information to support an argument that is being crafted by a senior attorney. Using indices and connections provided by legal materials, the junior lawyer must search through volumes of primary opinions and secondary legal commentary for the legal cases, legal theories, and statutory and regulatory citations to underpin an argument on a designated issue. Additionally, his[1] search must be completed within a certain time frame and is further constrained by the resources, such as legal materials, that are available. Obviously, exhaustive blind search is not viable because of the sheer volume of legal materials available. Thus, the junior associate must use heuristics to manage his research activities: researching new material based on approximate, though usually accurate, ideas of what's important to an argument, both in its details and in its overall quality. In



[1] Masculine pronouns should be read to encompass both males and females.

BankXX we model such research activities as a heuristically guided best-first search (e.g., [Nilsson, 1980]).

BankXX's analogue to the legal materials used by the junior lawyer are frames representing legal cases, legal theories, and other relevant constructs. BankXX searches through these using the classic method of heuristic best-first search: BankXX "expands" the current node to generate "successor" nodes; these are placed on a list of "open" nodes; a heuristic evaluation function is applied to them; and the best becomes the new current node, which is then mined for information it can yield for the evolving argument. Successor nodes are determined using the network representing legal knowledge and its interconnections. The evaluation functions capture knowledge at one of the three levels of abstraction described in (a), (b), and (c) above: domain knowledge, knowledge of what pieces are needed for an argument, and knowledge about what makes an argument good.

In summary, the BankXX system models the process of creating an argument through legal research as a heuristic search for relevant cases, theories, and other basic information. The information is incrementally analyzed and amalgamated into standard, desirable ingredients for an argument, such as citations to cases, applications of legal theories, and references to prototypical factual scenarios. The research reported here is an experiment to test the appropriateness of the search paradigm as a framework for researching—that is, for retrieving and mining—the knowledge needed to make legal arguments.

### 1.1. Perspective of this Research and Organization of the Paper

In this application, an argument is a collection of argument pieces, which represent fragments of arguments or pieces of legal knowledge that an advocate would ideally want to have to support his position. (Sections 4 and 6 contain a further description of what an argument is for our current purpose and an example.) The scope of the kinds of arguments that can be generated by the system is limited (and extended) by these argument pieces, which are modular and may be individually constructed or deleted from the system. Our current application is interpretative argument: to argue whether a bankruptcy plan satisfies the requirement that it be "proposed in good faith" (Section 2). At this point, it is for us a matter of future experiment with BankXX to determine what kinds of legal issues can be addressed and what kinds of arguments can be made within this straightforward framework. In this connection, we of course make no normative claim that the arguments produced by the system are the best (in some sense) or the ones that should be made. The task we have set ourselves is to implement a computer program that uses heuristic search in a network of legal materials to produce output that might be useful to an attorney framing a legal argument in a specific area of the law.

In one formulation of argument as search, the search space would be the space of all arguments, the start state would be an empty argument, and the search operators would represent ways to advance the argument. However, our current system models argument as the emerging by-product of the search and research that one might perform in a space of domain knowledge. We perform search in domain space rather than in argument space in part because we are motivated by the legal research activities of attorneys and partly because we are interested in the indexing fabric of the domain from a case-based reasoning perspective. See Table 1 for a capsule description of our approach to search in BankXX.

Table 1. Summary of the search model used by BankXX, with section references in this paper.

| |
|---|
| **Search States:** Set of nodes in a case-domain graph representing either a case at some level of abstraction or a legal theory (Section 3.1). |
| **Initial State:** (1) Problem situation or (2) user-specified node in the case-domain graph (Section 5.1). |
| **Operators on States:** Set of functions that trace a single link or a sequence of links in the case-domain graph. Here called "neighbor methods" (Section 3.3). |
| **Goal States:** None(Section 5.1). |
| **Termination Criteria:** (1) Empty open list, or (2) user-specified time bounds exceeded, or (3) user-specified space bounds exceeded (Section 5.1). |
| **Heuristic Evaluation:** Three linear evaluation functions at different levels of abstraction (Section 5.2). |

The remainder of the paper is organized as follows. In Section 2 we give some background on the cases, factors and legal theories of the bankruptcy "good faith" area. BankXX's representation of legal knowledge in this domain is described in Section 3. Section 4 describes how knowledge about argument, particularly argument pieces and argument dimensions, has been implemented in BankXX. Section 5 brings together the discussion of domain knowledge and argument knowledge by examining how search evaluation functions using either can guide search. A comparison of the opinion from an actual case and the argument generated by BankXX is made in Section 6. The paper's final two sections discuss how previous work on argument, legal knowledge representation, and search relates to this project, and give a brief summary.

## 2. The Bankruptcy "Good Faith" Domain

BankXX is instantiated in the area of bankruptcy law for individuals that is covered by Chapter 13 of United States bankruptcy law (11 U.S.C. §§ 1301-1330). Chapter 13 provides a means for an individual debtor to obtain relief from debts while keeping much of his property. By contrast, Chapter 7 (11 U.S.C. §§ 701-766) is based on liquidation of a debtor's assets to satisfy debts. Under Chapter 13 a debtor pays his creditors according to a court-approved plan that allocates 100% of his disposable income for a period of three to five years. Successful completion of the plan discharges the entire debt, regardless of the portion that is actually repaid.

There is potential for abuse of the debt-absolving power of Chapter 13. For example, a consumer could take out a large loan and spend the money with no intention of repaying it; a student could take out an educational loan and default on it without even trying to repay the loan. By declaring bankruptcy such a debtor would hope to get away with just repaying a small fraction of what is owed. One way the law is designed to prevent this and other abuse is by requiring that a repayment plan be "proposed in good faith" (11 U.S.C. § 1325(a)(3)).

Since Chapter 13 took effect as part of the Bankruptcy Reform Act in October, 1979, many cases have been litigated around the good faith issue. Evolving case law has elaborated on just what constitutes "good faith," a term left undefined in the original text of the law. Courts of Appeal for most of the federal circuits have articulated legal theories on the issue; to date the Supreme Court has not. The general approach taken by most courts has been to list a number of "factors" that a bankruptcy court should consider in making its decision. For example, one influential standard was articulated by the Eighth Circuit Court of Appeals in 1982 in *In re Estus*, 695 F.2d 311, 317 (8th Cir. 1982):

> ...in addition to the percentage of repayment to unsecured creditors, some of the factors that a court

may find meaningful in making its determination of good faith are:
(1) the amount of the proposed payments and the amount of the debtor's surplus;
(2) the debtor's employment history, ability to earn and likelihood of future increases in income;
(3) the probable or expected duration of the plan ....

Note that the *Estus* court leaves open the questions of whether these are all the factors that a bankruptcy court should consider and how are they to be applied: "We make no attempt to enumerate all relevant considerations since the factors and the weight they are to be given will vary with the facts and circumstances of the case." (*id.*) However, *Estus* does give special emphasis to one factor: the percentage of debt repaid to unsecured creditors, stating that "[a] low percentage proposal should cause the courts to look askance at the plan" (*id.*).

Thus, there are many "legal theories"—all based on a factor approach—in the corpus of cases addressing the "good faith" issue. Often the theories can be viewed as related or derived from each other. One theory might modify individual factors from another theory, or a theory might alter another theory's set of factors and their relative weightings. The theories have also been subject to changes in the bankruptcy code itself. In 1984 the Bankruptcy Amendments and Federal Judgeship Act added a requirement to section 1325 that 100% of a debtor's disposable income be used in the plan, eliminating the relevance of *Estus* factor (1) in subsequent cases.

## 3. Domain Knowledge in BankXX

### 3.1. The Case-Domain Graph

The case base in BankXX consists of a semantic network whose nodes represent cases and legal theories, and whose labeled links represent connections between the nodes. We refer to this network as the *case graph,* which consists of *case-graph nodes* together with labeled link edges.

Case-graph nodes represent legal cases (1) as factual situations, (2) as bundles of citations, (3) as stereotypical stories or scripts, (4) in terms of various legal factors, or (5) by the measure of their prototypicality. Legal theories are also represented as case-graph nodes.

The case graph is partitioned into *spaces,* similar to the partition of a blackboard application's working memory into spaces. Each space contains case-graph nodes that represent cases (or legal theories) according to a particular perspective that has proven useful to human legal reasoners (Figure 1). Nodes in the case graph are highly interconnected: in-space links connect objects within a space and cross-space links connect objects in different spaces. During search of the case graph, links are traversed by *neighbor methods,* operators that expand nodes in the graph by following either in-space links, cross-space links or a sequence of links. Traversing a link is tantamount to using the link label as an index.

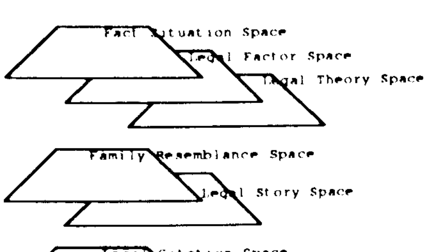


Figure 1. Spaces in the case graph.

We now describe each space of case-graph nodes, including its in-space indexing links and some cross-space links. Additional cross-space links are described in Section 3.2.

**Fact Situation Space.** Fact situation case nodes encode legal cases as sets of facts. Each situation is represented as a tree of frames implemented as CLOS instances. Examples of frames at this level describe the proposed plan and payments, the debt, the debtor's income, and generic information about the case. This level, which is the level at which cases are input, is the surface level of factual description. Cases at this level of representation are linked to each other through case citations (Figure 2).

**Legal Citation Space.** Citation case nodes encode cases as sets of citations: those that are cited by a case and those (later cases) that cite it. Citations include a citation signal that specifies the sense in which a case is cited (*see* [Ashley & Rissland, 1987]). Citation instances also include the West key numbers [West, 1993] indexing the topics addressed in the case. Citation instances are not linked to each other. They provide links to the citing case and the cited case in Fact Situation Space (Figure 2).

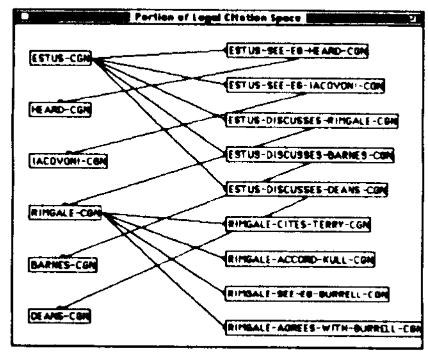


Figure 2. A small subgraph of the case graph, showing cases in Fact Situation Space (left side of figure) linked to citation nodes in Legal Citation Space (right side). "CGN" denotes a case-graph node.

**Legal Factor Space.** Legal cases can be represented in terms of their values on domain dependent factors [Rissland, Valcarce and Ashley, 1984], [Ashley, 1990]. Factors are derived features recognized by domain experts as strongly influencing a case's outcome. A factor compares cases as stronger or weaker with respect to the factor's perspective. In Legal Factor Space, a case is represented by a vector composed of the magnitudes of the case on each dimension, thereby representing a case as a point in an n-dimensional space.

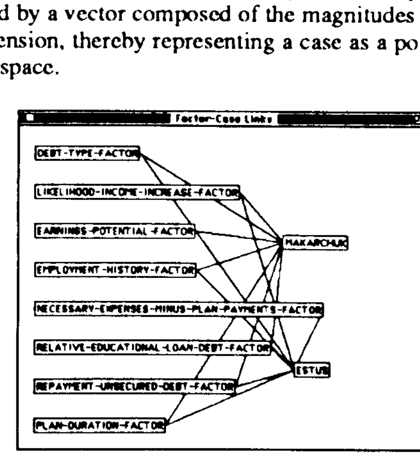


Figure 3. A small subset of the indexing links between domain factors and cases.

Examples of factors in BankXX are the percent of disposable income that is allocated to payments under the plan, the length of the proposed plan, and the portion of the debt that is attributable to educational loans. Using a HYPO-style analysis [Ashley, 1990], the system creates links between factors and the cases to which they apply (Figure 3).

**Legal Story Space.** In bankruptcy, as well as in other domains, cases often follow certain standard scripts or story lines. Two of the bankruptcy story prototypes used by BankXX are: (1) *the student loan story* — student incurs educational debts and soon after graduating files for bankruptcy protection from his educational loan creditors, and (2) *the dishonest debtor* — debtor commits fraud or some other offense, a judgment is entered against debtor, debtor files for bankruptcy.

BankXX does not link story prototypes to each other. Exploiting such links would require an understanding of how stories can be related and, ideally, an automated means to recognize them (e.g., plot units [Lehnert, 1981]).

**Family Resemblance Space.** We have begun to incorporate some of the research of Rosch, who proposed a model of the internal structure of categories that is captured in the family resemblance hypothesis: "the most prototypical members of categories are those with most attributes in common with other members of that category and are those with least attributes in common with other categories" [Rosch and Mervis, 1975, p.576]. While Rosch proposed the family resemblance hypothesis as a cognitive structural model, we are experimenting with family resemblance as an indexing and processing model. BankXX can calculate the degree of family resemblance of a case to a given set of cases and select the cases within that family having the greatest family resemblance. For instance, the system can calculate the family resemblance of all student loan cases, and find the most prototypical.

**Legal Theory Space.** Legal theories are represented as a list of factors (see the discussion of Legal Factor Space) that are necessary to determine how a theory applies to a case (Figure 4). Legal theory nodes are linked by pointers that describe the relationships between them (Figure 5), such as "overlaps with," "rejects," and "agrees with." In the next implementation, a legal theory node will specify how to combine the factors — for instance, via weighting — to apply the theory. Legal theories have been culled from opinions by hand.

```
(make-instance 'legal-theory
        :name 'ESTUS-THEORY
        :other-names '(FLYGARE-THEORY)
        :description "<omitted>"
        :factors '(surplus-factor
                percent-of-repayment-factor
                employment-history-factor
                earnings-potential-factor
                plan-duration-factor
                plan-accuracy-factor
                preferential-creditor-treatment-factor
                secured-claims-modified-factor
                debt-type-factor
                nondischarge-7-factor
                special-circumstances-factor
                frequency-relief-sought-factor
                motivation-sincerity-factor
                trustee-burden-factor)
        :factor-evaluation nil
        :domain-theories 'debt
        :view :majority
        :cases-promulgating '(estus)
        :cases-applying '(estus flygare makarchuk)
        :cases-rejecting nil
        :courts-adopting :8th-circuit            )
```

Figure 4. Representation of the Estus theory.

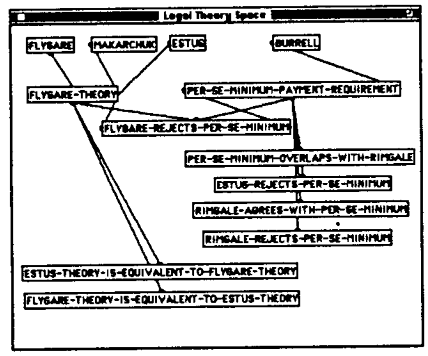


Figure 5. A small subgraph of the case graph, showing inter-theory links and links from theories to cases.

### 3.2. Cross-Space Case Links

In addition to the in-space links we have described, a variety of bi-directional, cross-space links exist. For instance, links exist between factors and legal theories that use those factors, and between story prototypes and cases instantiating them.

The current implementation contains 54 cases from the bankruptcy good faith domain, 70 citation links,[2] 21 inter-theory links of 8 types, 27 domain factors, 4 prototypical story scripts, and 12 argument pieces. The system runs on the Macintosh family of computers, and is written in Macintosh Common Lisp v.2.0 using CLOS. We have included graphing facilities that allow the user to examine the case network from a variety of indexing perspectives.

### 3.3. Neighbor Methods

The set of nodes to which a search algorithm may permissibly move in legal research is less constrained than in a classical search application like game-playing, where the moves that may legally take place define the possible successors. In legal research, the number of "legal moves" is extremely large due to the immense variety and volume of legal knowledge and the ways in which it can be manipulated. In the law, there are numbers of sources of compiled knowledge to aid the attorney or legal assistant in the task of finding relevant materials. For instance, Shepard's Citations gives inter-case and statute-case links for use in tracking down legal materials [Shepard's, 1992]. The West Publishing Company has developed a system of keys that index specific areas of legal practice. In addition, there are other, implicit links used in practice that are not reflected in standard materials: links that capture the fact that a case presents an instance of a typical, recurring fact situation, that is, a "story"; links between a case and the legal theory that is used to decide it, etc.

BankXX employs neighbor methods, which use links from the graph to generate possible nodes to examine in the search of the case-domain graph. Neighbor methods can (1) follow single links, (2) follow a series of links, or, in some cases, (3) create links dynamically. The neighbor methods contain the knowledge of how to move about in the case-domain graph. BankXX currently uses 12 neighbor methods to exploit the case-domain graph's high degree of interconnectedness.

Some examples of methods that follow a single link are: the method called *cases-theory*, which follows the links from a legal theory to the cases that have applied that legal theory; the method *cites*, which follows back pointers (from a case represented as a citation-bundle) to find those cases on which the given case relies;

[2]We have only partially entered the citations from each of the 54 cases in our case base.

and the method *cited-by*, which chases forward pointers to find cases that rely on the given case.

Neighbor methods of type (2) are similar to macro-operators [Fikes, Hart & Nilsson, 1972] in that they collapse a series of link tracings into a single one in order to perform a retrieval that has been recognized as useful by legal researchers. For example, the method *case-theory-neighbors* yields all the cases that have applied that theory. The more complicated neighbor method called *case-theory-theory-case* starts with a case, traces the link to the legal theories that are applied in that case, traces the links from each one of those theories to theories that have been cited favorably by that theory, and then traces the case graph edges back to the cases to those cases that have applied the favorably viewed legal theories. This method returns cases that have applied similar theories, and provides a useful collection of cases to examine in the next stage of the search (Figure 6).

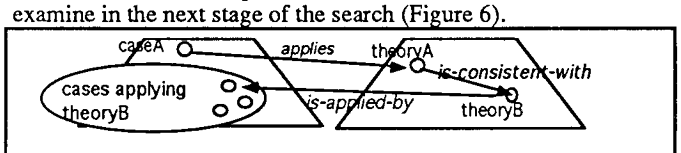


Figure 6. Tracing a series of links to perform a specific retrieval using the neighbor method *case-theory-theory-case.*

One neighbor method of type (3) uses a measure of case prototypicality to link prototypical cases with other cases that fall within the same prototype family. Some neighbor methods take the problem case as an functional parameter, so different paths can be traced through the case-graph from a given start node, depending on the current problem. For example, one neighbor method depends on the set of factors that apply to the problem in order to expand case-graph nodes in a context-specific manner.

## 4. Knowledge About Argument in BankXX

### 4.1. Some Building Blocks of Argument: Argument Pieces

We have chosen a simple representation of an "argument" for purposes of this implementation. In this application, an argument is a collection of argument pieces, which represent fragments of arguments or pieces of legal knowledge that an advocate would ideally like to have to support his position. The argument pieces represent building blocks of argument. We recognize that this idealization of argument does not reflect the logical and rhetorical connections between the various pieces of an argument, or the complexity of argument in general. Our immediate goal is to gather the information necessary to create a complete argument. The 12 argument pieces currently used in BankXX are:

- *cases decided for the current viewpoint*
- *best cases*[4]
- *leading cases*
- *cases sharing a large proportion of domain factors*
- *contrary cases decided for the opposing side*
- *contrary best cases for the opposing viewpoint*
- *family-resemblance-prototype:*[3]
- *supporting citations*
- *applicable legal theories*
- *nearly applicable supporting legal theories*
- *the factual prototype story category of the case*
- *factor analysis of the current problem*

For each argument piece there is a functional predicate that determines if a node can supply that useful piece of an argument.

[3]The cases decided with the desired viewpoint that have the highest family resemblance rating to the given case.

[4]Based on the definition of best case used in HYPO [Ashley, 1990].

Argument pieces also contain an object slot to store entities that satisfy its predicate. BankXX builds up their content incrementally (as its search proceeds) and the collection of all argument pieces is output to the user at the conclusion of BankXX's processing. There is no argument text generation facility within BankXX, however.

### 4.2. Argument Dimensions

Just as cases may be indexed and compared on the basis of domain factors [Rissland, Valcarce & Ashley, 1984], [Ashley, 1990], so may arguments be compared on the basis of "argument factors." Argument factors capture dimensions along which the quality of arguments may be compared and contrasted. They can aid the system in identifying the best arguments (e.g., by sorting arguments according to a partial order based on the factors that apply to an argument). The third type of evaluation function we have experimented with in BankXX is based on these factors; this function is also used to evaluate the final argument output by BankXX.

Eight *argument factors* are currently implemented in BankXX:

(1) *centrality-of-best-cases,*
(2) *centrality-of-theory,*
(3) *win-record-of-theory,*
(4) *win-record-of-theory-for-factual-prototypes,*
(5) *strength-of-best-case-analogies,*
(6) *factual-prototypicality-strength,*
(7) *strength-of-citations,* and
(8) *equally-on-point-cases.*

Briefly, the meaning of the argument factors is as follows: (1) *Centrality-of-best-cases* assesses centrality of the best cases retrieved by determining how often those cases have been cited in other cases in the case base. (2) *Centrality-of-theory* determines how often the theory has been used/invoked in a case or compared to a theory used in a case. (3) *Win-record-of-theory* determines the proportion of times that the theory has been used that it has resulted in a winning argument according to the decision rendered in each case. (4) *Win-record-of-theory-for-factual-prototypes* determines the proportion of cases in which the theory has been successfully used in a case whose facts follow a recognized, stereotypical pattern (e.g., a *student-loan* case.) (5) *Strength-of-best-case-analogies* is implemented currently in terms of the average number of legal factors that are in common to both the best cases cited in the argument and the current problem. (6) *Factual-prototypicality-strength* computes a normalized family resemblance rating for the current problem, to determine how prototypical it is for cases of this prototype (e.g., student-loan). (7) *Strength-of-citations* gives a measure of how often the cases mentioned in the supporting citations argument piece point to leading cases or best cases. (8) *Equally-on-point-cases* measures the proportion of best cases for which there are no equally on-point cases for the opposing side that share the same subset of dimensions.

## 5. Traversing the Case Graph to Support an Argument

### 5.1. Search and Control Flow

The goal of BankXX is to examine the nodes of the case-domain graph to provide information that may be used to support a legal argument. This examination is performed using the mechanisms of classical heuristic search.

In general, state-space search is defined by a triple: (initial state, set of operators on states, set of goal states). The search performed by BankXX differs from the usual applications in two ways: the complexity of node expansions through the neighbor methods and the absence of well-defined goal states. Neighbor methods are described in Section 3.3. We do not include goal states in our model because of the difficulties inherent in defining an "argument goal" in a way that is consistent with our informal understanding of how humans develop and evaluate legal arguments. It is hard to say in general that an argument does or does not meet some plausible persuasive or rhetorical goal, or even that one has completed the supporting research.

In best-first search, an evaluation function is also used to guide the exploration of the state-space [Barr et al., 1981]. In BankXX the initial state is a user-supplied problem situation, which is represented using the same set of hierarchical frames used to represent a case as a collection of facts. Alternatively, a start node may be specified by the user. For instance, a user may wish to select a leading case in the area, like the *Estus* case, if he knows of one, in order to concentrate search initially in a particular region of the space.

BankXX begins its processing by analyzing the problem situation for applicable domain factors and computing a claim lattice, which sorts the cases that have some of the same factors at work as the current problem. The best and most on-point cases are identified. These provide potential new nodes to be explored. The system continues by performing iterative, best-first search. The neighbors of the current node are generated using the neighbor methods. The "best" node on the open list—the one with the maximum value under one of BankXX's evaluation function—is identified and is then examined by each of the argument pieces in turn in order to determine if it can be used as a component of the argument. This cycle continues until the search exceeds a user-specified time or space bound, or until the open list is empty. At the conclusion of the search, the argument is assessed in terms of the argument dimensions and BankXX outputs the argument to the user in a template structured by the argument pieces. In this way the information needed to build up the various argument pieces and ultimately the overall argument is acquired incrementally during the search (Figure 7).

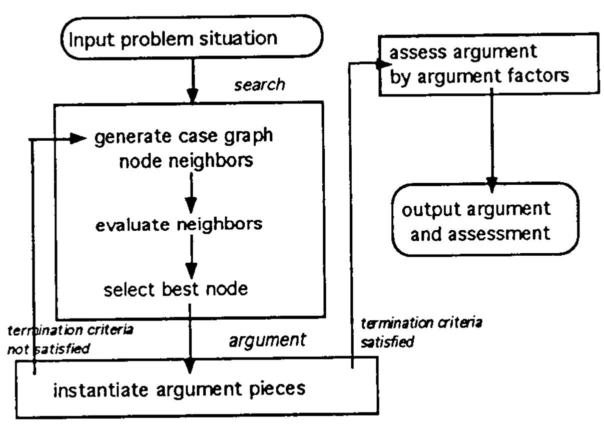


Figure 7. Control flow of the BankXX system.

### 5.2. Heuristic Evaluation Functions

We have experimented thus far with three different types of evaluation functions. They differ in the level of abstraction that they use to evaluate nodes in the case-domain graph. All of the evaluation functions are simple linear functions. They form a progression of increasingly more informed evaluation methods, whose considerations range from (1) only the type of information encoded in a node to (2) the contribution of the node to the standard argument pieces and (3) the incremental impact of a node on the overall state of the evolving argument. Briefly, the functions at the three levels of abstraction are:

(1) **The domain level.** The form of this evaluation function is $\sum w_i f_i(c)$. The $f_i$ check the *type* (e.g., legal-case, legal theory) of the node, $c$. The $w_i$ are non-negative, scalar-valued weights in each of the three evaluation function forms. This function is of the general form:

$$w_1\, type\text{-}pred_1(c) + w_2\, type\text{-}pred_2(c) + \ldots + w_n\, type\text{-}pred_n(c)$$

Essentially the question asked by this evaluation function is: "How well will this node contribute information of a type known to be useful to argument?"

(2) **The argument piece level.** The form of this function is $\sum w_i f_i(c, a)$, where $c$ is the current node and $a$ is the current state of the argument. Each $f_i$ computes if a particular *argument piece* is fillable by the current node and if that argument piece has *not* already been completely filled: if so, $f_i$ returns 1; else, 0. This intermediate-level evaluation function prevents BankXX from wasting computing resources by unnecessary bolstering of parts of the argument that are already well-established. It is of the general form:

$$w_1\, arg\text{-}piece\text{-}pred_1(c,a) + w_2\, arg\text{-}piece\text{-}pred_2(c,a) + \ldots + w_n\, arg\text{-}piece\text{-}pred_n(c,a)$$

Essentially the question asked by this evaluation function is: "Can the particular domain knowledge contained in this node be used to complete one of the desired components of an argument that has not already been completely filled?"

(3) **The argument level.** The form of this evaluation function is $\sum w_i f_i(c, a, a^*)$, where $a^*$ is the argument that would result from incorporating the knowledge in node $c$ into the current argument $a$. The $f_i$ compare the values along each of the *argument dimensions* applied to the current argument $a$ with those of the argument $a^*$. It is of the general form:

$$w_1\, arg\text{-}dim\text{-}fcn_1(c,a,a^*) + w_2\, arg\text{-}dim\text{-}fcn_2(c,a,a^*) + \ldots + w_n\, arg\text{-}dim\text{-}fcn_n(c,a,a^*)$$

Essentially the question asked by this evaluation function is: "Can the domain knowledge contained in this node improve the quality of the argument?"

### 6. An Example: *In Re Ali*

*In Re Ali*, 33 B.R. 890 (Bankr. D. Kan. 1983) is a case involving a Kansas couple, Majid and Hasiba Ali, who proposed a repayment plan that would pay off a loan secured by their car, but would pay nothing to unsecured creditors. The University of Kansas, where Majid Ali had been a student and to which he owed five thousand dollars in student loans, objected, arguing that the plan had not been proposed in good faith.

In ruling on the good faith question, the bankruptcy court applied the legal theory enunciated by the Court of Appeals in its circuit in the case *Flygare v. Boulden*, 709 F.2d 1344 (10th Cir. 1983). *Flygare* adopts verbatim the list of factors from *Estus*. In the *Ali* decision each of these factors was considered in turn and the judge ultimately decided that the Alis filed their plan in good faith.

As an experiment, we input the facts of the *Ali* case to BankXX as a new problem case to determine to what extent BankXX identified the elements of argumentation used by the court in the actual *Ali* decision. *Ali* is in the system's case base, but we have removed it and its linkages so it can be used *de novo* as a problem for the system. For this example, we asked BankXX to start its search from the well-known *Estus* case. For this example we used an evaluation function at the domain level, type (1) above.

BankXX identified a number of the important elements used in the actual decision (Figure 8). For example, it identified the theory in *In re Kitchens*, 702 F.2d 885 (11th Cir. 1983), as an applicable legal theory. The search performed by BankXX proceeded from the *Estus* case by following a link to the *Estus* theory, and from there it discovered related theories, including the *Kitchens* theory. Deciding that the theory was relevant, BankXX added it to the applicable theories argument piece. The actual *Ali* decision specifically states (at 894) that it follows the rule set forth in *Kitchens*.

From the *Kitchens* theory, BankXX found the *Kitchens* case itself and added it to its list of cases to cite in the argument. But *Kitchens* was decided in favor of the creditors. So BankXX identifies it as a "contrary case." One important issue in *Kitchens* was that plans calling for payments lasting less than 3 years will be subject to special scrutiny. The *Ali* problem case is differentiated by the fact that it proposes payments lasting 3 years. This difference is mentioned in the actual *Ali* decision.

BankXX also finds the *Flygare* theory on which the actual *Ali* decision is based. It identifies *Flygare* as containing a nearly applicable legal theory, and finds and adds the *Flygare* case to the list of leading cases to cite. *Flygare* was found by following the path from *Estus* to the *Estus* theory, to the identical *Flygare* theory, and then to the *Flygare* case itself.

In addition to legal theories, BankXX identifies the *Ali* problem case as belonging to the student loan prototype. *Estus* is a student loan case, so BankXX examines the student loan prototype to see if it is relevant. Finding that it is, it adds the prototype to the *factual-prototype-story* argument piece.

**LEADING-CITED-CASES:**
(<FLYGARE-><IACOVONI><RIMGALE><DEANS>
<ESTUS>)
**NEARLY-APPLICABLE-THEORIES:**
(<ALL-THE-FACTS-ANDCIRCUMSTANCES>
<RIMGALE-THEORY> <FLYGARE-THEORY>
<MAKARCHUK-PRINCIPAL-PURPOSE-STUDENT-LOAN-DISCHARGE><ESTUS-THEORY>...)
**APPLICABLE-THEORIES:**
(<KITCHENS-KULL-THEORY><OLD-BANKRUPTCY-ACT-GOOD-FAITH-DEFINITION><PER-SE-MINIMUM-PAYMENT-REQUIREMENT>)
**CURRENT-FACTOR-ANALYSIS:**
<ALI-FACTOR-ANALYSIS>
**CONTRARY-CASES:**
(<SANDERS><SELLERS><HEARD><IACOVONI>
<ESTUS>)
**SAME-SIDE-CASES:**
(<FLYGARE><SCHYMA><VALENTINE><RIMGALE>
<BARNES><DEANS>)
**SUPPORTING-CITATIONS:**
(<RIMGALE-CITES-TERRY><RIMGALE-ACCORD-KULL><RIMGALE-SEE-EG-BURRELL><RIMGALE-AGREES-WITH-BURRELL>)
**CONTRARY-BEST-CASES:** (<IACOVONI>)
**SUPPORTING-BEST-CASES:**
(<FLYGARE><SCHYMA><VALENTINE><RIMGALE>)
**FACTUAL-PROTOTYPE-FAMILY-RESEM.:** NIL
**DOMAIN-FACTOR-OVERLAP:** NIL
**FACTUAL-PROTOTYPE-STORY:**
(<STUDENT-LOAN-FACTUAL-PROTOTYPE>)

Figure 8. Argument output by BankXX for the *Ali* case.

## 7. Related Research

We do not discuss generally here either argument or legal argument, which are treated well and at length elsewhere (e.g., [Perelman & Olbrechts-Tyteca, 1969], [Toulmin, 1958], [Levi, 1949]), or argument modeled through other means than search ([McCarty & Sridharan, 1982], [Sycara, 1989], [Alvarado, 1990]). In addition, our present goal is not to provide a formal, logical model of legal argument. We refer the reader to [Gordon, 1991] and [Prakken, 1993] for excellent discussions.

Several researchers have addressed aspects of argument as search. [Bhatnagar, 1989] treat an argument as a search for a causal model that supports a given proposition. Bhatnagar uses a variant of A* search to create models that satisfy argument goals, in which it assumed that probability values may be computed for the validity of supported propositions given a particular model. While we also view argument creation as theory construction [Rissland & Skalak, 1991], we believe that such a probabilistic approach may be difficult to apply in a domain as "weak" as law.

Branting's GREBE system [Branting, 1991] uses structured representations of the explanations for legal decisions supplied in the opinions of legal cases. It uses heuristic A* search for one aspect of argument creation: retrieval of a precedent that best explains a problem case. Best-first search is performed in a space consisting of all mappings from a problem case to these structured representations of precedent cases. Thus, GREBE's use of A* search is not in the same search space as that of BankXX, but search is used to the same end—to retrieve relevant cases.

While we do not rely directly on research using artificial neural networks for information retrieval or on related, massively parallel techniques, the flavor of some of this work is similar to our approach. In particular, Rose and Belew's SCALIR [1991] is a hybrid symbolic and sub-symbolic system that uses a network of legal knowledge, including Shepard's links and West's key number taxonomy links, through which numerical activation is spread to perform retrieval.

This research also reflects the knowledge organization of earlier work in conceptual legal retrieval (e.g., [Hafner, 1987a, 1987b], [Bing, 1987], [Dick, 1987]) that relies on a graph of diverse legal entities and concepts with labeled links capturing influences and taxonomic information.

For the notions of analyzing representation of cases in terms of important domain factors or "dimensions" [Rissland, Valcarce & Ashley, 1984], the construction of "claim lattices," which partially order retrieved cases by dimensions, and the selection of best and most on-point cases, we rely on ideas developed in HYPO [Ashley, 1990].

## 8. Summary and Contributions

BankXX is an experiment in bottom-up argument generation. It is an attempt to make computational our experience of how one builds an argument through legal research, particularly in areas where one doesn't have the expertise to provide a top-down view of a desirable argument. We have chosen to implement the research task as best-first search in a network of legal knowledge in order to determine whether search, a general, so-called "weak" method, can make adequate decisions as to what pieces of potentially supporting knowledge to investigate. While our previous research suggests that some combination of bottom-up and top-down is necessary for creating arguments [Skalak & Rissland, 1992], the BankXX system provides another data point in how to control systems that generate legal arguments. Future work will evaluate in detail the success of BankXX in generating useful support for arguments, but preliminary comparisons with actual opinions have been encouraging.


## Acknowledgments

This work was supported in part by the Air Force Office of Sponsored Research under contract 90-0359. For assistance we thank Matt Cornell and Kate Sanders, and Elaine Whitlock and Barbara Fell-Johnson of the Hampshire County Law Library, Northampton, Massachusetts.

## References


Alvarado, S. J. (1990). *Understanding Editorial Text: A Computer Model of Argument Comprehension*. Boston, MA: Kluwer Academic Publishers.

Ashley, K. D. (1990). *Modeling Legal Argument: Reasoning with Cases and Hypotheticals*. Cambridge, MA: M.I.T. Press.

Ashley, K. D. & Rissland, E. L. (1987). But, See, Accord: Generating Blue Book Citations in HYPO. *Proceedings, The First International Conference on AI and Law*, 67-74. Boston, MA. New York, NY: Association for Computing Machinery.

Barr, A., Feigenbaum, E. A. & Cohen, P. (1981). *The Handbook of Artificial Intelligence*. Reading, MA: Addison-Wesley.

Bhatnagar, R. K. (1989). *Construction of Preferred Causal Hypotheses for Reasoning with Uncertain Knowledge*. Ph.D. Thesis, University of Maryland, College Park, MD.

Bing, J. (1987). Designing Text Retrieval Systems for "Conceptual Searching". *The First International Conference on Artificial Intelligence and Law*, pp. 43-51. Boston, MA. New York, NY: Association for Computing Machinery.

Branting, L. K. (1991). Building Explanations from Rules and Structured Cases. *International Journal of Man-Machine Studies*, 34, 797-837.

Dick, J. P. (1987). Conceptual Retrieval and Case Law. *Proceedings, The First International Conference on Artificial Intelligence and Law*, 106-114. Boston, MA. New York, NY: Association for Computing Machinery.

Erman, L. D., Hayes-Roth, F., Lesser, V. R. & Reddy, D. R. (1980). The HEARSAY-II Speech Understanding System: Integrating Knowledge to Resolve Uncertainty. *Computing Surveys*, 12.

Fikes, R. E., Hart, P. & Nilsson, N. J. (1972). Learning and executing generalized robot plans. *Artificial Intelligence*, 3, 251-288.

Gordon, T. F. (1991). An abductive theory of legal issues. *International Journal of Man-Machine Studies*, 35, 95-118.

Hafner, C. D. (1987). *An Information Retrieval System Based on a Computer Model of Legal Knowledge*. Ph.D. Thesis, University of Michigan. Republished by UMI Research Press, Ann Arbor, MI (1981).

Hafner, C. D. (1987). Conceptual Organization of Case Law Knowledge Bases. *Proceedings, The First International Conference on Artificial Intelligence and Law*, 35-42. Boston, MA. New York, NY: Association for Computing Machinery.

Lehnert, W. (1981). Plot Units and Narrative Summarization. *Cognitive Science*, 5(4).

Levi, E. H. (1949). *An Introduction to Legal Reasoning*. Chicago: University of Chicago Press.

McCarty, L. T. & Sridharan, N. S. (1982). *A Computational Theory of Legal Argument* (LRP-TR-13). Laboratory for Computer Science Research, Rutgers University, New Brunswick, NJ.

Nilsson, N. (1980). *Principles of Artificial Intelligence*. Palo Alto, CA: Tioga Press.

Perelman, C. & Olbrechts-Tyteca, L. (1969). *The New Rhetoric: A Treatise on Argumentation*. Notre Dame, Indiana: University of Notre Dame Press.

Prakken, H. (1993). *Logical Tools for Modelling Legal Argument*. Ph.D. Thesis, Vrije Universiteit te Amsterdam, Amsterdam, The Netherlands.

Rissland, E. L. & Skalak, D. B. (1991). CABARET: Rule Interpretation in a Hybrid Architecture. *International Journal of Man-Machine Studies*, 34, 839-887.

Rissland, E. L., Valcarce, E. M. & Ashley, K. D. (1984). Explaining and Arguing with Examples. *AAAI-84, Proceedings of the National Conference on Artificial Intelligence*. Austin, TX. American Association for Artificial Intelligence.

Rosch, E. & Mervis, C. B. (1975). Family Resemblances: Studies in the Internal Structure of Categories. *Cognitive Psychology*, 7, 573-605.

Rose, D. E. & Belew, R. K. (1991). A Connectionist and Symbolic Hybrid for Improving Legal Research. *International Journal of Man-Machine Studies*, 35, 1-33.

Shepard's. (1992). *Shepard's Federal Citations*. Colorado Springs, CO: Shepard's/McGraw-Hill.

Skalak, D. B. & Rissland, E. L. (1992). Arguments and Cases: An Inevitable Intertwining. *Artificial Intelligence and Law: An International Journal*, 1, 3-48.

Stucky, B. K. (1986). *Understanding Legal Argument*. Counselor Project Technical Memorandum 13. Department of Computer Science, University of Massachusetts, Amherst, MA.

Sycara, K. P. (1989). Argumentation: Planning Other Agents' Plans. *Proceedings, Eleventh International Joint Conference on Artificial Intelligence*, 517-523. Detroit, MI. International Joint Conferences on Artificial Intelligence.

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

West. (1993). St. Paul, MN: West Publishing Co.